



Alignment d'images pour analyser les mécanismes de déformation des matériaux

HAMZA AZIZI

MASTER BIG DATA INTELLIGENCE ARTIFICIELLE

Juin 2023 - Novembre 2023

D

Tuteur Académique :

TRISTAN CASENAVE
UNIVERSITÉ PARIS-DAUPHINE - PSL

Tuteurs Laboratoire :
FRÉDÉRIC PENNERATH - LORIA
BRIEUC CONAN-GUEZ - LORIA
LYDIA BOUDJELOUD - LORIA
ANTOINE GUITTON - LEM3
BENOÎT BEAUSIR - LEM3
SATYAVEER DHINWAL - LEM3
VINCENT TAUPIN - LEM3

Résumé

Cette étude a pour objectif d'analyser les déformations des matériaux à l'échelle microscopique en élaborant une méthode d'alignement d'images. Les résultats démontrent l'efficacité de cette approche dans l'exploration des déformations, ouvrant ainsi des perspectives quant à l'application de solutions d'intelligence artificielle pour prédire les déformations d'un matériau. Ce rapport offre une analyse approfondie du processus de recherche, des méthodes employées, des résultats obtenus, et propose des pistes pour des développements futurs.

Table des matières

1 Présentation	4
1.1 Présentation du projet et des laboratoires	4
1.2 Objectifs et contributions	4
2 Introduction	5
2.1 Notions fondamentales	5
2.1.1 Microstructure d'un matériau	5
2.1.2 Angles d'Euler	6
2.1.3 Construction des cartes EBSD	7
2.1.4 Comportement des grains face aux contraintes	8
2.1.5 Analyse des déformations d'un matériau	8
2.2 Méthodes DIC (Digital Image Correlation)	9
2.2.1 Approche Locale	9
2.2.2 Approche Globale	10
2.3 Présentation du projet	10
2.3.1 Problématique	10
2.3.2 Inspiration issue de l'Imagerie médicale	11
2.3.3 Idée générale de la méthode proposée	11
2.3.4 Méthodes de transformations géométriques	12
3 Méthodologie	14
3.1 Méthode d'alignement de deux images	14
3.1.1 Objectif de l'étude	14
3.1.2 Description de la méthode	14
3.1.3 Le critère d'alignement	15
3.2 Déformation de l'image d'arrivée	16
3.2.1 Transformation par quadrilatére	16
3.2.2 Interpolation bilinéaire	16
3.3 Déformation de la grille d'arrivée	18
3.3.1 Problème d'optimisation	18
3.3.2 Descente de gradient	19
3.3.3 ADAM	20
3.4 Récapitulatif des phases du processus de la méthode proposée	21
3.5 Avantages de la méthode	23
3.6 Inconvénients et limitations	24
3.7 Améliorations envisageables	24
3.8 Perspectives futures	25
4 Expériences et résultats	25
4.1 Laminage	25
4.1.1 Position initiale de la grille d'arrivée	26
4.1.2 Variation de la taille de grille	31
4.2 Traction uniaxiale	35
4.2.1 Variation de la taille de grille	35
4.2.2 Alignement de la première image avec les suivantes	38
4.3 Conclusion	41

Table des figures

1	Réseau spatial d'atomes	5
2	Représentation en 2D d'un matériau au niveau des grains et des réseaux atomiques de chaque grain	6
3	Rotations et angles d'Euler	6
4	(a) : Analyse dans le SEM et formation du diagramme de diffraction, (b) : Indexation du diagramme, (c) : Construction de la carte EBSD	7
5	Les contraintes examinées dans ce projet	8
6	Déformation des grains d'un matériau soumis à la traction uniaxiale	8
7	Analyse des déplacements entre deux images	9
8	Comparaison des deux approches du DIC[2]	9
9	Transformation géométrique par quadrilatère.	12
10	Exemple d'un matériau avant et après l'application de la contrainte	14
11	Processus itératif de la méthode.	15
12	Interpolation Bilinéaire	17
13	Exemple de quadrilatère non convexe	17
14	Exemple d'application de la méthode	18
15	Visualisation des itérations du modèle.	22
16	Évolution du critère d'alignement en fonction des époques	23
17	Initialisation des grilles - Test 1.	26
18	Résultats de la convergence - Test 1.	27
19	Résultats de la convergence - Test 2.	27
20	Résultats de la convergence - Test 3.	28
21	Résultats de la convergence - Test 4.	29
22	Le quadrilatère qui n'a pas été bien aligné pour les tests 2 et 4	29
23	Le quadrilatère qui n'a pas été bien aligné pour les tests 2 et 4 avec les angles de désorientation et pas les angles d'Euler	30
24	Résultats de la convergence - Test 5.	30
25	Résultats de la convergence - Test 6.	31
26	Résultats de la convergence - Test 7.	31
27	Résultats de la convergence - Test 8.	32
28	Résultats de la convergence - Test 9.	32
29	Résultats de la convergence - Test 11.	33
30	Résultats de la convergence - Test 12.	34
31	Résultats de la convergence - Test 13.	34
32	Essais de traction uniaxiale sur inconel 625	35
33	Initialisation des grilles - Test 14.	36
34	Résultats de la convergence - Test 14.	36
35	Résultats de la convergence - Test 15.	37
36	Résultats de la convergence - Test 16.	37
37	Résultats de la convergence - Test 17.	38
38	Résultats de la convergence - Test 18.	39
39	Résultats de la convergence - Test 19.	39
40	Vecteurs de déplacement.	40

1 Présentation

Ce travail de recherche a été entrepris dans le cadre d'un stage de fin d'études de master 2, réalisé conjointement au sein de deux laboratoires de recherche renommés, le **LORIA** (Laboratoire Lorrain de Recherche en Informatique et ses Applications) et le **LEM3** (Laboratoire d'Etude des Microstructures et de Mécanique des Matériaux).

1.1 Présentation du projet et des laboratoires

Le sujet de stage s'inscrit dans le projet **MAMIENOVA** (Machine leArning for MIcromEc-hanics : a NOVel Approach), projet financé par l'ANR (Agence Nationale de la Recherche). Un projet dont l'objectif est d'exploiter des solutions d'intelligence artificielle en science des matériaux.

L'équipe de MAMIE NOVA est composée de chercheurs de 3 laboratoires français : **LEM3**, **UMET**, et **LORIA**.

Le **LORIA**, Laboratoire lorrain de Recherche en Informatique et ses Applications, est une Unité Mixte de Recherche (UMR 7503), commune à plusieurs établissements : le CNRS, l'Université de Lorraine et Inria. Depuis sa création en 1997, le LORIA a pour mission la recherche fondamentale et appliquée en sciences informatiques.

Le **LEM3**, Laboratoire d'étude des microstructures, de la mécanique et des sciences des matériaux, également appelé LEM3, est un laboratoire de recherche français situé à Metz et est une entité CNRS (UMR 7239) existant depuis le 1er janvier 2011. Il est sous la tutelle des Arts et Métiers ParisTech, de l'Université de Lorraine et de l'ENIM. Le LEM3 se distingue par ses recherches approfondies en mécanique des matériaux et en études des microstructures.

L'**UMET** (Unité Matériaux et Transformations) est un laboratoire de recherche créée en 2010 lors de la refonte des UMR lilloises antérieures, spécialisé en science des matériaux.

Liée à l'Université de Lille, à l'École nationale supérieure de chimie de Lille ainsi qu'au CNRS.

1.2 Objectifs et contributions

Les objectifs spécifiques de ce travail de recherche consistent à mettre en œuvre une technique d'optimisation dans la vision par ordinateur, adaptée à l'étude des déformations des matériaux. Cette démarche vise à approfondir notre compréhension des mécanismes de déformation, ouvrant ainsi la voie à des avancées significatives dans le domaine de la science des matériaux.

Les contributions attendues de ce projet consistent en la création d'un algorithme d'alignement d'images à l'aide d'outils d'apprentissage automatique. Ce travail représente également une étape préparatoire essentielle pour établir un modèle d'apprentissage automatique en vue de prédire les déformations d'un matériau.



2 Introduction

Les sciences des matériaux constituent un domaine de recherche visant à comprendre les matériaux utilisés dans une variété d'applications, de l'aéronautique à l'industrie automobile. L'une des questions fondamentales qui préoccupent les chercheurs dans ce domaine est la manière dont la structure microscopique des matériaux influence leurs propriétés et leur comportement. Comprendre ces relations est essentiel pour le développement de matériaux aux performances améliorées.

2.1 Notions fondamentales

2.1.1 Microstructure d'un matériau

Les matériaux métalliques sont souvent interprétés en fonction de leur structure interne, appelée **microstructure**.

Un matériau est composé de **grains**, et ces grains sont des réseaux spatiaux d'atomes ordonnés. Chaque grain, souvent invisible à l'œil nu, représente une région où les atomes interagissent de manière cohérente.

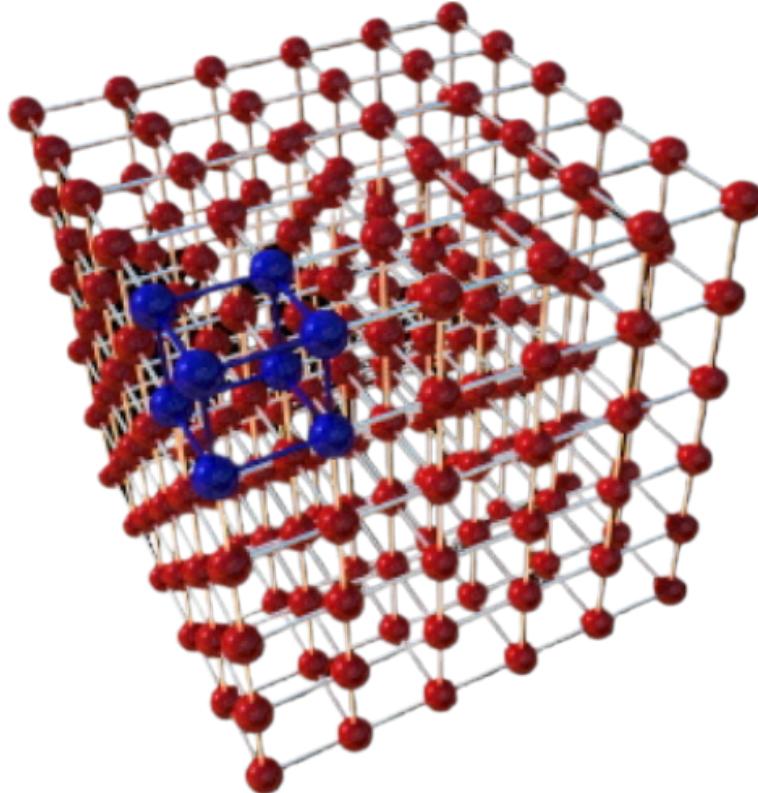


FIGURE 1 – Réseau spatial d'atomes

Les matériaux se divisent en deux grandes familles : les **monocristallins**, constitués d'un seul grain, et les **polycristallins**, constitués de plusieurs grains. Pour cette étude, nous nous intéressons à la famille des matériaux polycristallins.

Chaque grain est orienté dans l'espace, et pour définir cette orientation spatiale, on utilise les angles d'Euler.

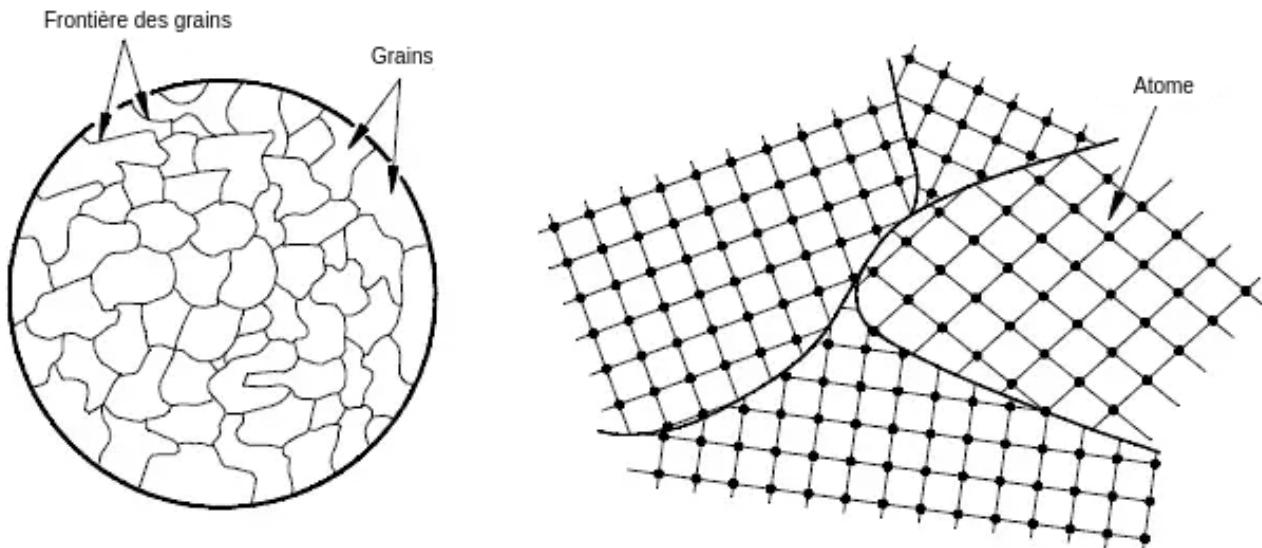


FIGURE 2 – Représentation en 2D d'un matériau au niveau des grains et des réseaux atomiques de chaque grain

2.1.2 Angles d'Euler

En mécanique et en mathématiques, les **angles d'Euler**[1] sont des angles introduits par Leonhard Euler (1707-1783) qui permettent de décrire l'orientation spatiale d'un objet par rapport à un système de référence (x , y , z).

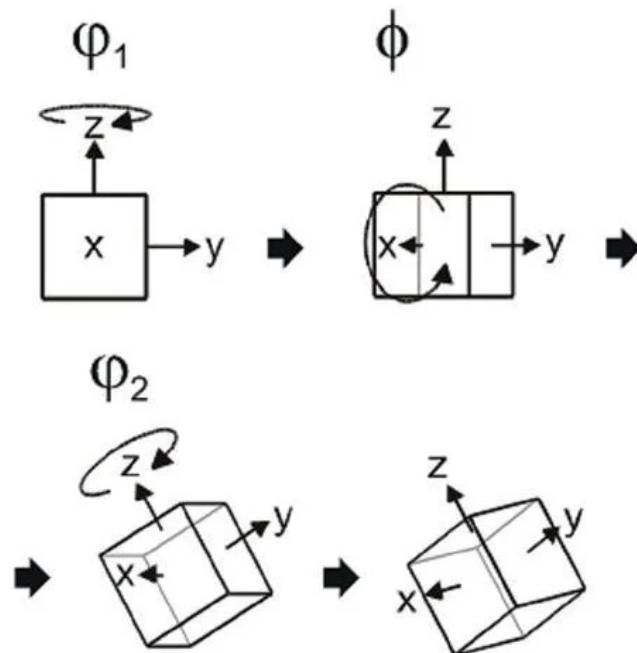


FIGURE 3 – Rotations et angles d'Euler

Lorsque l'on dispose des valeurs des angles d'Euler en tous points de l'échantillon du matériau, il est possible de les représenter sous forme d'information colorimétrique, indiquant l'orientation spatiale de chaque grain dans une image RGB à 3 canaux (où chaque canal prend une valeur d'un angle d'Euler), représentant le matériau en deux dimensions. Ces représentations visuelles sont appelées des cartes EBSD.

2.1.3 Construction des cartes EBSD

Les **cartes EBSD** sont des images illustratives des grains composant le matériau, avec une information colorimétrique renseignant sur l'orientation de chaque grain. Ces images sont produites par un microscope électronique à balayage.

EBSD : L'EBSD[6] est une technique d'analyse cristallographique locale. Elle permet de relier la cristallographie à la microstructure de l'échantillon.

SEM : Dans le contexte de la microscopie, SEM fait référence au Microscope Électronique à Balayage (Scanning Electron Microscope en anglais), un type de microscope qui utilise un faisceau d'électrons pour créer une image de la surface d'un échantillon.

Étapes de construction des cartes EBSD :

1. **Préparation de l'échantillon :** L'échantillon doit être préparé de manière à ce que sa surface soit suffisamment plane et propre pour permettre une interaction efficace avec le faisceau d'électrons (polissage).
2. **Analyse dans le SEM :** L'échantillon est ensuite placé dans un SEM, où un faisceau d'électrons est dirigé vers la surface de l'échantillon incliné. Les électrons interagissent avec l'échantillon et sont rétrodiffusés[3].
3. **Formation du diagramme de diffraction :** Les électrons rétrodiffusés forment un diagramme de diffraction sur un écran fluorescent. Ce diagramme est capturé par une caméra [3].
4. **Indexation du diagramme :** Le diagramme de diffraction est ensuite analysé pour déterminer l'orientation du point analysé [3].
5. **Construction de la carte EBSD :** En répétant ce processus pour différents points sur l'échantillon, une carte EBSD complète peut être construite. Cette carte montre l'orientation cristalline spatiale à différents points de l'échantillon.

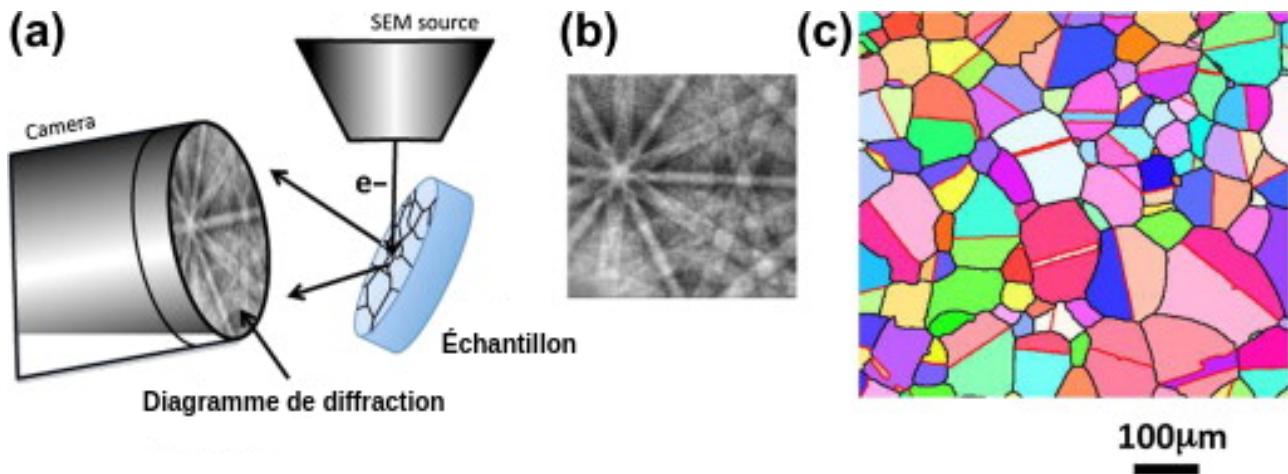


FIGURE 4 – (a) : Analyse dans le SEM et formation du diagramme de diffraction, (b) : Indexation du diagramme, (c) : Construction de la carte EBSD

Ces cartes EBSD constituent les données avec lesquelles nous allons travailler afin d'analyser la déformation d'un matériau.

2.1.4 Comportement des grains face aux contraintes

Les tests de contraintes sur les matériaux sont couramment utilisés pour évaluer leur comportement. Parmi les nombreux tests de contraintes disponibles, ce projet se concentre spécifiquement sur l'analyse de matériaux soumis aux processus de laminage et de traction uniaxiale.

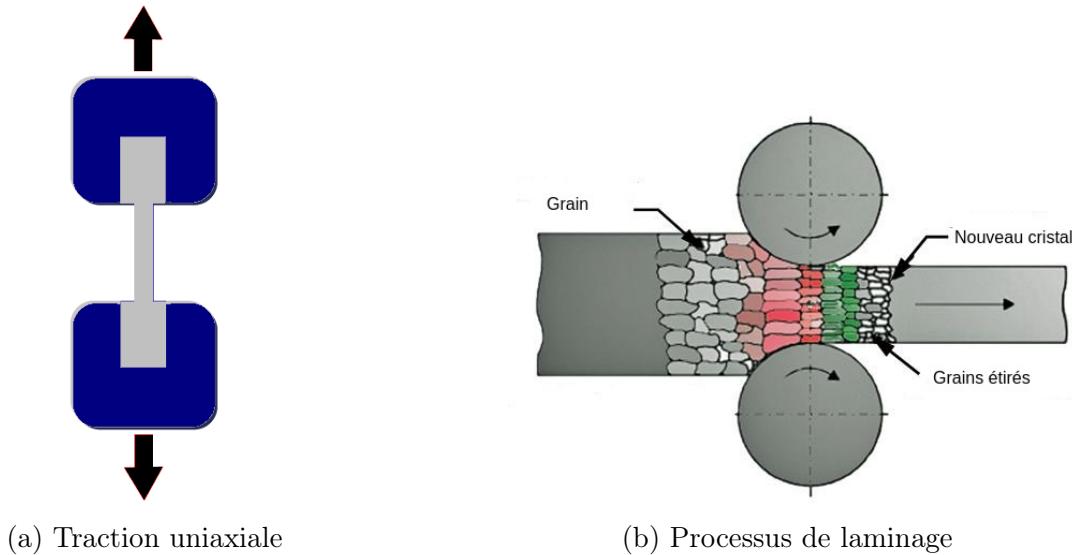


FIGURE 5 – Les contraintes examinées dans ce projet

Si un matériau subit une déformation en raison d'une contrainte qui lui est appliquée, comme c'est le cas avec le laminage dans la figure (5-b), cela se traduit par la déformation et le changement d'orientation spatiale des grains qui le composent. En cas de contrainte significative, on peut observer l'apparition de nouveaux cristaux suite à la cassure d'un grain.

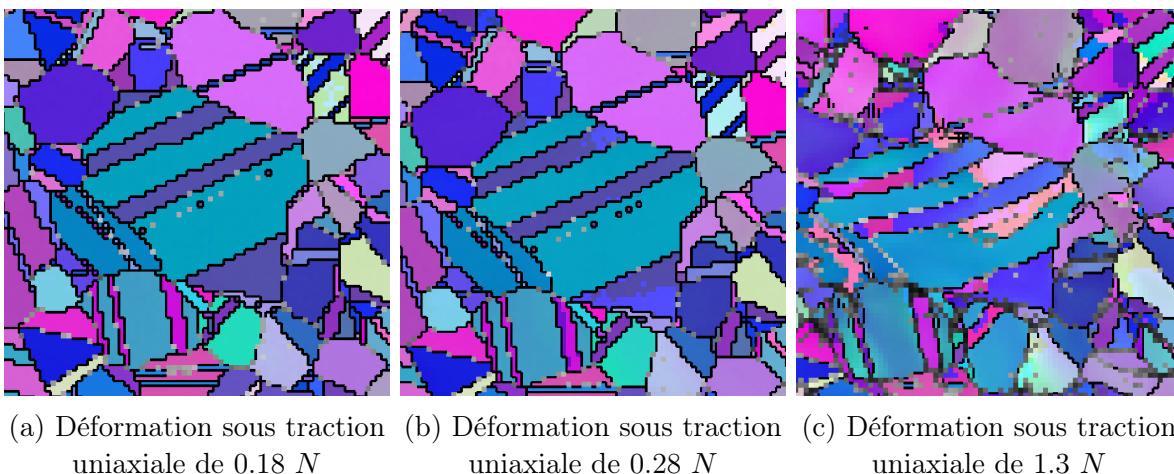


FIGURE 6 – Déformation des grains d'un matériau soumis à la traction uniaxiale

2.1.5 Analyse des déformations d'un matériau

L'analyse des déformations dans les matériaux revient à déterminer le champ de déplacement. Le champ de déplacement représente la manière dont les points d'un matériau se déplacent et se déforment sous l'influence de contrainte, permettant ainsi de quantifier les variations de forme et de position des éléments constitutifs.

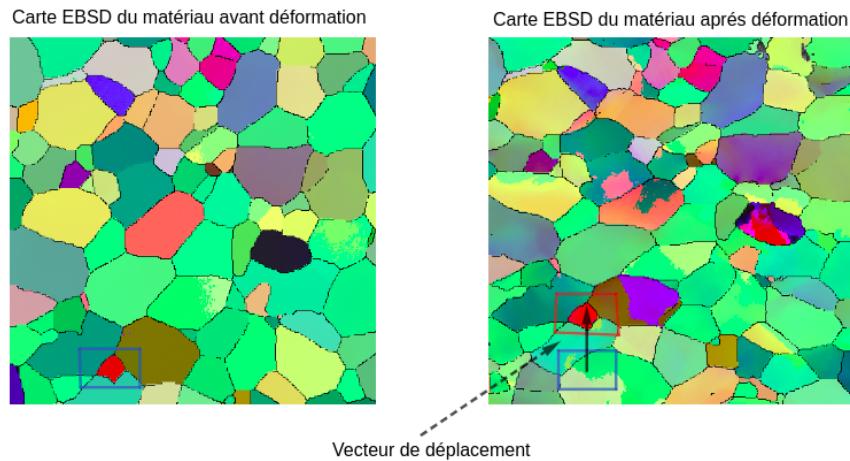


FIGURE 7 – Analyse des déplacements entre deux images

Comme le montre la figure (7), à gauche, nous avons la carte EBSD du matériau avant déformation, et à droite, la carte EBSD après déformation. Un vecteur de déplacement sera calculé pour chaque point de l'échantillon afin de former l'ensemble du champ de déplacement.

2.2 Méthodes DIC (Digital Image Correlation)

La corrélation d'images numériques (Digital Image Correlation ou DIC en anglais)[4][6] est une méthode optique 2D ou 3D qui permet de mesurer les déplacements entre deux images. Elle est de plus en plus employée en sciences des matériaux pour déterminer le champ de déplacement.

Un champ de déplacement est une représentation spatiale qui quantifie comment la forme, la taille et la géométrie d'un objet ou d'une région varient, il est utilisé pour suivre les changements et les mouvements d'un objet.

Il existe deux approches pour la corrélation d'images numériques principales : locale et globale

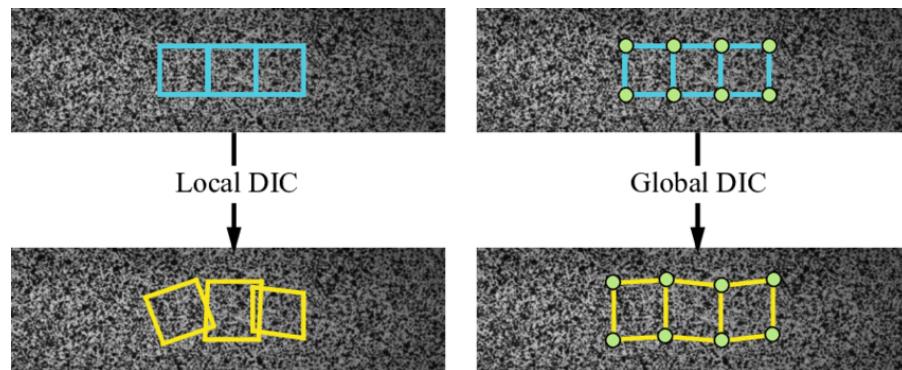


FIGURE 8 – Comparaison des deux approches du DIC[2]

2.2.1 Approche Locale

Développée initialement aux États-Unis à partir des années 1980. La corrélation d'images numériques avec une approche locale consiste à diviser l'image de référence en imagettes. Pour chaque imagette, on cherche sa partie correspondante dans l'image déformée qui maximise une fonction de corrélation croisée calculée entre l'imagette et la partie choisie de l'image déformée.

Cette approche est dite "locale" car elle traite chaque imagette de manière indépendante. En conséquence, elle fournit un vecteur de déplacement uniquement pour le centre de chaque imagette, sans fournir un champ de vecteurs pour tous les pixels de l'image.

Avantages :

- **Haute résolution locale** : L'approche locale offre une résolution élevée pour chaque imagette, permettant de détecter des déformations fines au niveau des régions d'intérêt.
- **Calculs plus rapides** : Étant donné que l'analyse est effectuée sur des imagettes individuelles, les calculs sont généralement plus rapides que dans une approche globale qui traite l'ensemble de l'image.
- **Sensibilité aux détails locaux** : L'approche locale est idéale pour l'analyse des variations locales de contraste et la détection de déformations fines.

Inconvénients :

- **Information limitée** : L'approche ne fournit des informations que pour les centres des imagettes, ne donnant pas un champ de déplacement complet pour tous les pixels de l'image.
- **Possibilité de discontinuités** : En raison de l'analyse indépendante des imagettes, des discontinuités peuvent apparaître lorsque les déformations entre deux imagettes voisines ne sont pas prises en compte.

2.2.2 Approche Globale

Développée en France dans les années 2000, l'approche globale consiste à diviser l'image de référence en une grille ou en une série de régions, tout en superposant une grille similaire sur l'image déformée. En optimisant une fonction globale, les correspondances entre les régions de l'image de référence et de l'image déformée sont recherchées à une plus grande échelle. Contrairement à l'approche locale, l'approche globale permet d'obtenir une vue d'ensemble des déformations sur l'ensemble de l'image. Voici les avantages et les inconvénients des méthodes DIC globales :

Avantages :

- **Vue d'ensemble** : L'approche globale permet d'obtenir une vue d'ensemble des déformations à une échelle plus large de l'image ou de la scène.
- **Moins sensible au bruit local** : Elle peut être plus robuste face aux variations locales de contraste ou au bruit, car elle intègre des informations sur une plus grande région.
- **Idéale pour des déformations globales** : Elle est particulièrement adaptée pour l'analyse des déformations globales dans des applications telles que la surveillance de structures.

Inconvénients :

- **Baisse de la résolution** : L'approche globale a généralement une résolution plus faible, ce qui peut limiter la capacité à détecter des déformations locales ou fines.
- **Calculs plus lents** : En raison de l'analyse d'une plus grande surface, les calculs peuvent être plus lents et nécessiter davantage de ressources informatiques.
- **Moins adaptée aux détails locaux** : Elle peut ne pas être aussi efficace pour la détection de déformations fines ou de petits détails dans l'image.

2.3 Présentation du projet

2.3.1 Problématique

Un matériau soumis à une contrainte subit une déformation, se traduisant par la modification des grains qui le composent. Nous disposons de deux images des grains du matériau avant et après l'application de la contrainte, et nous souhaitons analyser cette évolution. Pour ce faire,

nous envisageons de développer une méthode d'alignement d'images, c'est-à-dire une méthode permettant de définir une transformation géométrique à appliquer à l'image du matériau après l'application de la contrainte, de manière à ce qu'elle se superpose de manière optimale avec l'image du matériau avant l'application de la contrainte. En trouvant une telle transformation géométrique, il serait possible de quantifier les vecteurs de déplacement en tous points du matériau, afin d'analyser l'évolution des grains sous contrainte.

En principe, il serait préférable de transformer l'image du matériau avant la contrainte vers celle après la contrainte. Cependant, nous avons décidé de le faire dans l'autre sens parce que c'est plus facile à coder. Nous expliquerons en détail la raison de ce choix lors de la description du fonctionnement de l'algorithme.

Cette méthode se veut une alternative aux approches existantes de la Corrélation d'Images Numériques (DIC), mettant l'accent sur la rapidité d'exécution. Elle bénéficie des avantages offerts par les tenseurs en PyTorch, permettant d'accélérer les opérations de calcul.

2.3.2 Inspiration issue de l'Imagerie médicale

En réfléchissant à notre problématique, il devient clair que la solution que nous cherchons à développer est similaire aux solutions présentes dans un domaine bien établi de l'imagerie médicale : l'alignement d'images(Image Registration[6] en anglais).

Dans le domaine de l'imagerie médicale[6], notamment en ce qui concerne les images par résonance magnétique (IRM), l'objectif est de parfaitement aligner des images. Par exemple, pour la segmentation[14] et l'analyse des tissus, l'alignement des images facilite la délimitation des structures anatomiques, c'est-à-dire l'identification des organes et des tissus. Cette analogie est particulièrement pertinente pour notre travail, car les matériaux que nous étudions comportent des grains dont les mouvements et les déformations présentent des défis similaires à ceux des tissus en imagerie médicale. L'alignement d'images en imagerie médicale repose sur des principes qui s'inspirent de la nécessité de corriger les différences de position, d'échelle, de rotation et de déformation entre des images acquises à des moments distincts par exemple suivre les tissus qui bougent et se déforment. Une des approches clés pour réaliser cet alignement consiste à appliquer des transformations géométriques aux images et un critère d'alignement est optimisé.

Les transformations géométriques utilisées en imagerie médicale sont mises en œuvre à l'aide d'algorithmes qui estimate les paramètres de transformation nécessaires pour aligner les images de manière optimale. Ils sont souvent basés sur la recherche de correspondances entre des caractéristiques significatives dans les images, telles que les descripteurs SIFT[14], et l'estimation des paramètres de transformation qui minimisent la divergence entre les images.

2.3.3 Idée générale de la méthode proposée

Nous avons développé un algorithme pour étudier les déformations d'un matériau en combinant la corrélation d'images numériques (DIC) avec des transformations géométriques couramment utilisées en imagerie médicale.

L'objectif de cet algorithme est de déterminer une transformation géométrique à appliquer à l'image du matériau après l'application de la contrainte, créant ainsi une version déformée qui se superpose correctement avec l'image du matériau avant l'application de la contrainte.

Pour ce faire, nous plaçons deux grilles, une sur chaque image, afin de définir la transformation géométrique. En superposant les grilles, nous divisons chaque image en quadrillatères (les cellules de la grille). Chaque quadrilatère dans l'image du matériau après l'application de la contrainte est associé à son homologue dans l'image du matériau avant l'application la contrainte, établissant ainsi une transformation géométrique locale. Cette transformation locale est ensuite appliquée aux coordonnées des pixels du quadrilatère de l'image du matériau après l'application de la contrainte, les plaçant dans un quadrilatère similaire à celui de l'image du

matériau avant l'application de la contrainte. Chaque paire de quadrilatères des deux grilles définit une transformation locale.

Après avoir appliqué ces transformations géométriques locales à chaque partie de l'image du matériau après l'application de la contrainte, nous regroupons ces parties transformées pour obtenir la version déformée totale de l'image du matériau après l'application de la contrainte.

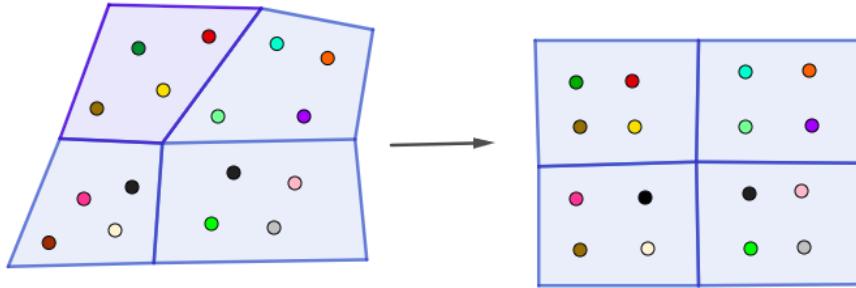


FIGURE 9 – Transformation géométrique par quadrilatère.

Figure 9 : À gauche, une grille divise l'image du matériau après l'application de la contrainte en parties délimitées par des quadrilatères. Les pixels (couleurs de la carte EBSD) sont situées à l'intérieur de ces quadrilatères. À droite, une grille uniforme où nous allons placer les couleurs correspondantes pour chaque partie de l'image du matériau après l'application de la contrainte.

Ainsi, nous produisons une version déformée de l'image du matériau après l'application de la contrainte, que nous souhaitons voir se superposer correctement avec l'image du matériau avant l'application de la contrainte.

2.3.4 Méthodes de transformations géométriques

Pour aligner un quadrilatère dans un autre, plusieurs méthodes peuvent être appliquées. Nous pouvons en citer deux bien connues :

Transformation perspective La transformation perspective[18][19], également connue sous le nom de projection perspective, est une méthode de transformation géométrique qui permet de modéliser les distorsions qui se produisent lorsqu'on regarde un objet tridimensionnel depuis un point de vue différent.

La transformation perspective est décrite par la formule suivante :

$$\begin{bmatrix} x' \\ y' \\ w' \end{bmatrix} = \begin{bmatrix} P_{11} & P_{12} & P_{13} \\ P_{21} & P_{22} & P_{23} \\ P_{31} & P_{32} & P_{33} \end{bmatrix} \begin{bmatrix} x \\ y \\ w \end{bmatrix}$$

où :

- $\begin{bmatrix} x \\ y \\ w \end{bmatrix}$ sont les coordonnées homogènes du point d'origine. Le terme w est un facteur d'échelle qui permet de représenter des coordonnées en 3D ; $w = 1$ en 2D.
- $\begin{bmatrix} x' \\ y' \\ w' \end{bmatrix}$ sont les coordonnées homogènes du point transformé.
- $[P_{ij}]$ est la matrice de transformation perspective qui dépend des paramètres de la caméra ou du point de vue. $P_{33} = 1$ en 2D.

Avantages :

- **Modélisation réaliste de la perspective :** Les transformations perspectives permettent de modéliser les distorsions de la perspective, ce qui est essentiel pour représenter des vues en 3D ou pour corriger des distorsions dans des images prises avec une perspective.
- **Applications 3D :** Les transformations perspectives sont indispensables pour des applications en trois dimensions, telles que la reconstruction 3D, la réalité augmentée et la vision par ordinateur.

Inconvénients :

- **Calculs plus complexes :** Les transformations perspective sont mathématiquement plus complexes que les transformations affines, ce qui peut entraîner une charge de calcul plus importante.
- **Perte de certaines propriétés géométriques :** Contrairement aux transformations affines, les transformations perspectives ne préservent pas les parallèles, les rapports de longueurs et les angles relatifs entre les points. Elles introduisent des distorsions qui peuvent affecter les propriétés géométriques des objets.
- **Discontinuité entre les quadrillatères :** La méthode que nous souhaitons développer est basée sur l'application de transformations locales pour chaque quadrilatère de la grille, puis l'assemblage des quadrillatères transformés. Cette transformation perspective ne garantit pas la continuité entre les bords des quadrillatères transformés.

Transformation affine La transformation affine[17] est une méthode de transformation géométrique utilisée pour des transformations linéaires telles que la translation, la rotation, la mise à l'échelle et le cisaillement.

Il est important de noter que la transformation affine ne permet pas d'aligner un quadrilatère sur un quadrilatère, mais plutôt un triangle sur un triangle.

Par conséquent, si l'on souhaite appliquer cette transformation à un quadrilatère, il faudrait d'abord le diviser en deux triangles, puis appliquer la transformation affine séparément à chaque triangle.

La transformation affine est décrite par la formule suivante :

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix}$$

où :

- $\begin{bmatrix} x \\ y \end{bmatrix}$ sont les coordonnées du point d'origine.
- $\begin{bmatrix} x' \\ y' \end{bmatrix}$ sont les coordonnées du point transformé.
- $\begin{bmatrix} a & b \\ c & d \end{bmatrix}$ est la matrice de transformation qui représente la rotation, la mise à l'échelle et le cisaillement.
- $\begin{bmatrix} t_x \\ t_y \end{bmatrix}$ est le vecteur de translation.

Avantages :

- **Conservation des propriétés géométriques :** Les transformations affines préservent les parallèles, les rapports de longueurs et les angles relatifs entre les points. Cela garantit que les formes géométriques de base (comme les quadrillatères) restent intactes.
- **Calculs plus simples :** Les transformations affines sont mathématiquement plus simples à manipuler, ce qui les rend plus efficaces en termes de calcul.

- **Idéales pour l'alignement initial :** Les transformations affines sont couramment utilisées pour corriger des différences de position, d'orientation et d'échelle entre les images. Elles sont idéales pour l'alignement initial avant d'appliquer des déformations plus complexes.

Inconvénients :

- **Difficulté à coder pour les quadrilatères :** Puisque la transformation affine aligne un triangle sur un triangle, pour l'adapter aux quadrilatères, il faut l'appliquer deux fois sur les deux triangles du quadrilatère quand on le divise.

3 Méthodologie

Dans cette section, nous expliquerons en détail la méthode utilisée pour aligner deux images et comment évaluer cet alignement. Cette méthodologie sera divisée en sous-sections pour une compréhension plus claire.

3.1 Méthode d'alignement de deux images

3.1.1 Objectif de l'étude

L'objectif principal est de mettre au point une méthode d'alignement pour deux images. Dans ce contexte, l'alignement désigne le processus d'ajustement et de mise en correspondance des deux images de manière à les superposer de manière optimale.

Cette méthode sera appliquée aux images microscopiques d'un matériau soumis à une contrainte afin d'analyser la déformation des grains qui le composent suite à l'application de la contrainte. Cette déformation sera analysée à l'aide des vecteurs de déplacement.

On désignera par 'image de départ' l'image du matériau avant l'application de la contrainte et par 'image d'arrivée' l'image du matériau après l'application de la contrainte.

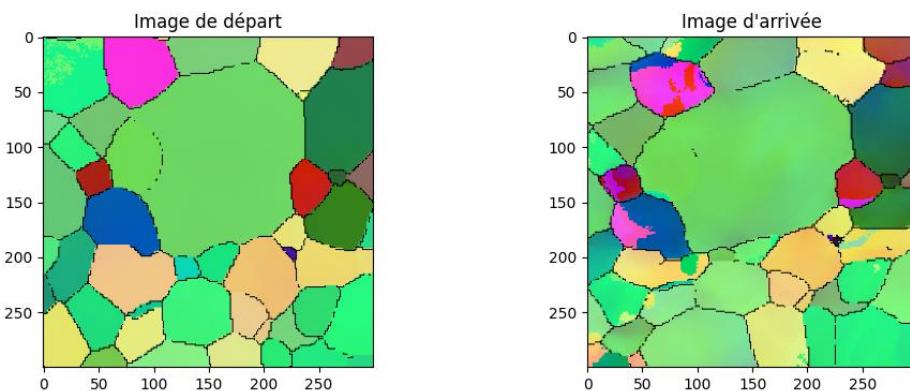


FIGURE 10 – Exemple d'un matériau avant et après l'application de la contrainte

3.1.2 Description de la méthode

La méthode vise à déterminer la transformation géométrique pour un alignement optimal entre l'image de départ et l'image d'arrivée.

Pour réaliser cette transformation géométrique, nous utilisons un processus itératif basé sur **l'utilisation de deux grilles pour définir une transformation géométrique globale** : une grille de départ placée sur l'image de départ et une grille d'arrivée placée sur l'image

d'arrivée. Cette paire de grilles définit une transformation géométrique globale à appliquer sur l'image d'arrivée. Cette transformation globale résulte de l'assemblage de transformations locales définies par chaque paire de quadrilatères, où un quadrilatère de la grille de départ correspond à un quadrilatère de la grille d'arrivée, déterminant ainsi une transformation locale appliquée aux pixels contenus dans le quadrilatère d'arrivée, les transformant indépendamment des autres pixels contenus dans d'autres quadrilatères.

Le processus itératif repose également sur l'**optimisation d'un critère d'alignement** qui guide la déformation de la grille d'arrivée, responsable de la déformation de l'image d'arrivée. La transformation est paramétrée par la grille d'arrivée, celle-ci étant la seule à se déformer au cours des itérations. La grille de départ reste toujours fixe. À chaque itération, le paramètre change, et chaque paramètre définit une transformation géométrique associée, appliquée à l'image d'arrivée, créant ainsi une image d'arrivée déformée.

Une fois l'image d'arrivée déformée produite, un critère d'alignement est calculé entre l'image de départ et l'image d'arrivée déformée. Plus le critère d'alignement est faible, meilleure est la superposition des deux images. Ce critère guide la déformation de la grille d'arrivée en calculant les gradients du critère par rapport au paramètre. Le paramètre est un vecteur de dimension nombre de noeuds de la grille d'arrivée en ligne et 2 en colonne, c'est les coordonnées des noeuds de la grille d'arrivée.

Après le calcul des gradients, le paramètre est mis à jour en ajustant l'emplacement des noeuds de la grille d'arrivée. Cette nouvelle grille d'arrivée sera utilisée dans l'itération suivante pour définir la prochaine transformation géométrique à appliquer à l'image d'arrivée.

Ce processus est répété jusqu'à convergence.

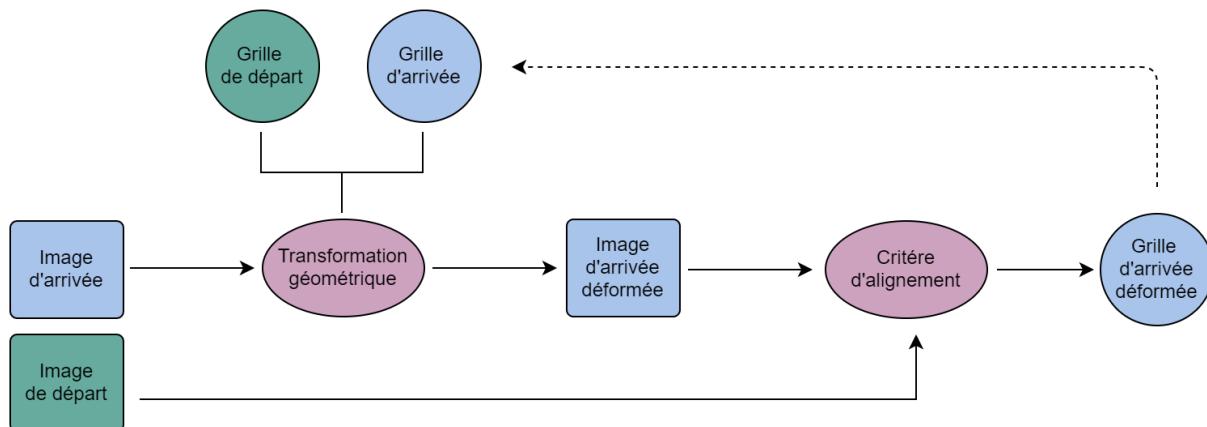


FIGURE 11 – Processus itératif de la méthode.

3.1.3 Le critère d'alignement

La déformation de la grille d'arrivée est obtenue par un procédé itératif, basé sur l'optimisation d'un critère numérique d'alignement entre l'image de départ et l'image d'arrivée déformée, un critère mesurant les différences entre les pixels de l'image de départ et les pixels déplacés de l'image d'arrivée.

Notez bien que le critère n'est calculé que sur l'intérieur des grilles, et non sur l'ensemble de l'image.

Nous avons initialement choisi d'utiliser l'erreur quadratique moyenne (MSE) classique. Cependant, cette approche a tendance à poser des problèmes aux bords de l'image si on a des sorties de zone, lorsque la grille d'arrivée évolue en dehors de l'image d'arrivée d'origine au fil

des époques. Pour résoudre ce problème, nous avons modifié ce critère en excluant les pixels situés en dehors de l'image d'arrivée.

$$L(w) = \frac{1}{\sum_{p \in P_D} \mathbb{1}_{\mathbb{T}_\omega[p] \in \mathbb{P}_{IA}}} \sum_{p \in P_D} \|\mathbb{I}_D[p] - \mathbb{I}_A[\mathbb{T}_\omega[p]]\|^2 \mathbb{1}_{\mathbb{T}_\omega[p] \in \mathbb{P}_{IA}} \quad (1)$$

- $L(w)$: critère d'alignement
- w : coordonnées des noeuds de la grille d'arrivée $w \in \mathbb{R}^{N*2}$ avec N le nombre des noeuds.
- \mathbb{P}_D : ensemble des points entiers à l'intérieur de la grille de l'image de départ.
- \mathbb{P}_A : ensemble des points entiers à l'intérieur de la grille de l'image d'arrivée.
- \mathbb{P}_{IA} : ensemble des points entiers de l'image d'arrivée.
- $\mathbb{I}_D : \mathbb{N}^2 \mapsto \mathbb{R}^3$ fonction qui à un point p associe la couleur en p de l'image de départ.
- $\mathbb{I}_A : \mathbb{N}^2 \mapsto \mathbb{R}^3$ fonction qui à un point p associe la couleur en p de l'image d'arrivée.
 \mathbb{I}_A prend la couleur noire ([0, 0, 0] en RGB) si $p \notin \mathbb{P}_{IA}$
- $\mathbb{T}_\omega : \mathbb{P}_D \mapsto \mathbb{P}_A$ transformation qui, à un point p de l'image de départ, attribue les coordonnées du point transformé $\mathbb{T}_\omega(p)$ dans l'image d'arrivée.

3.2 Déformation de l'image d'arrivée

Dans notre méthodologie d'alignement optimal de l'image de départ avec l'image d'arrivée, notre objectif est de déformer l'image d'arrivée de manière à obtenir la meilleure superposition possible avec l'image de départ. Dans cette section, nous explorerons en détail le processus de déformation de l'image d'arrivée, en utilisant les deux grilles d'arrivée et de départ.

3.2.1 Transformation par quadrilatére

Pour déformer l'image d'arrivée, nous effectuons une déformation individuelle de chaque partie de l'image, soutenue par un quadrilatère de la grille d'arrivée. La transformation paramétrique est globale et est donc spécifiée par une transformation par quadrilatère.

À chaque quadrilatère de la grille de départ est associé un quadrilatère de la grille d'arrivée. Cette correspondance définit une transformation associée à ce couple de quadrilatères, et notre objectif est de trouver une technique qui aligne un quadrilatère de départ vers son correspondant dans l'image d'arrivée.

3.2.2 Interpolation bilinéaire

Théorie L'interpolation bilinéaire[4][5] dans l'imagerie est une technique utilisée pour estimer la valeur d'un pixel de l'image de départ dans l'image d'arrivée, en fonction des valeurs des 4 pixels voisins dans lesquels il est inséré. On obtient ainsi une image transformée qui est une version déformée de l'image d'arrivée, pour nous les pixels voisins en fonction desquels on estime les coordonnées transformées sont les sommets des quadrilatères.

Application de la méthode Pour illustrer le concept, considérons un point P situé dans le quadrilatère à gauche de la figure. Nous cherchons à déterminer les coordonnées de son image transformée, le point Q , qui se trouve dans le quadrilatère à droite.

Théoriquement, si on note $u = \frac{AJ}{AD}$ et $v = \frac{AI}{AB}$, pour chercher Q , nous devons trouver les points Q_1 et Q_2 qui vérifient :

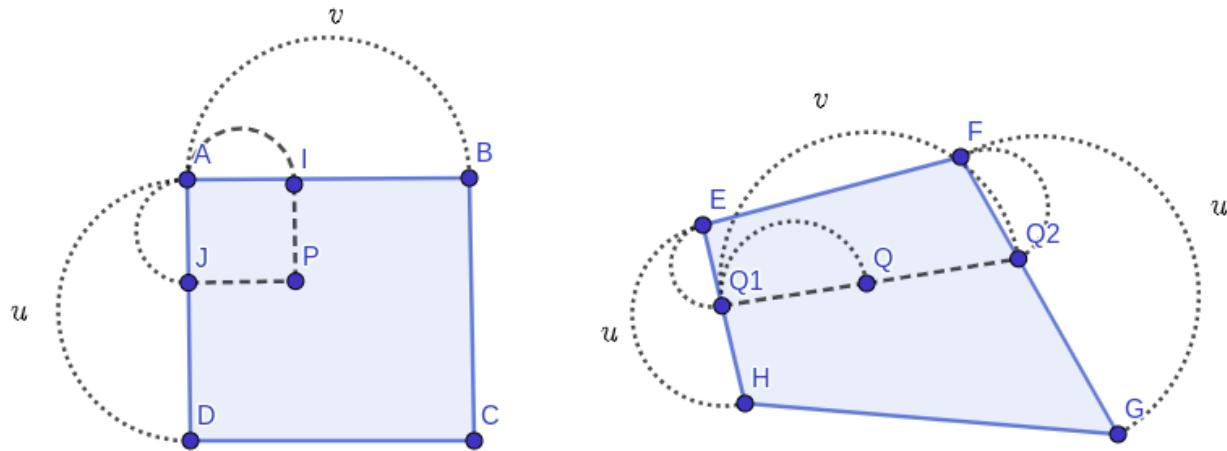


FIGURE 12 – Interpolation Bilinéaire

À gauche : Quadrilatère de la grille de départ.

À droite : Quadrilatère de la grille d'arrivée.

$$\begin{cases} \frac{EQ_1}{EH} = \frac{FQ_2}{FG} = u \\ \frac{Q_1Q}{Q_1Q_2} = v \end{cases}$$

Analytiquement, nous calculons $Q_1 = E + u \cdot EH$ et $Q_2 = F + u \cdot FG$. En conséquence, $Q = Q_1 + v \cdot Q_1Q_2$.

L'interpolation bilinéaire nous définit une transformation par quadrilatère qui sera appliquée aux pixels contenus dans le quadrilatère d'arrivée. La transformation paramétrique globale est continue car les transformations locales par quadrilatères se raccordent bien sur les arêtes. Chaque transformation locale est définie correctement seulement si les quadrilatères sont convexes.

Un quadrilatère non convexe est un quadrilatère caractérisé par au moins un angle interne de plus de 180 degrés. Cela se traduit par la présence d'une encoche ou d'un creux dans le quadrilatère, où une partie de son contour se courbe vers l'intérieur. Dans le cas de tels quadrilatères non convexes, un quadrilatère n'est plus aligné sur un autre quadrilatère, de sorte que, visuellement, la transformation n'est plus correcte, car cette méthode suppose toujours que les quadrilatères sont convexes pour maintenir la continuité de la transformation.

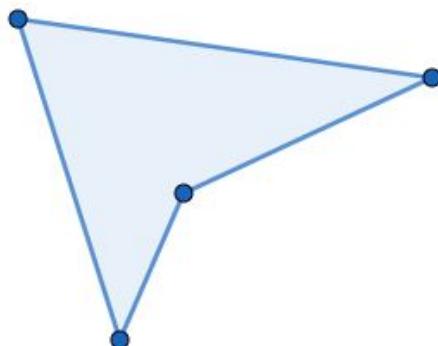


FIGURE 13 – Exemple de quadrilatère non convexe

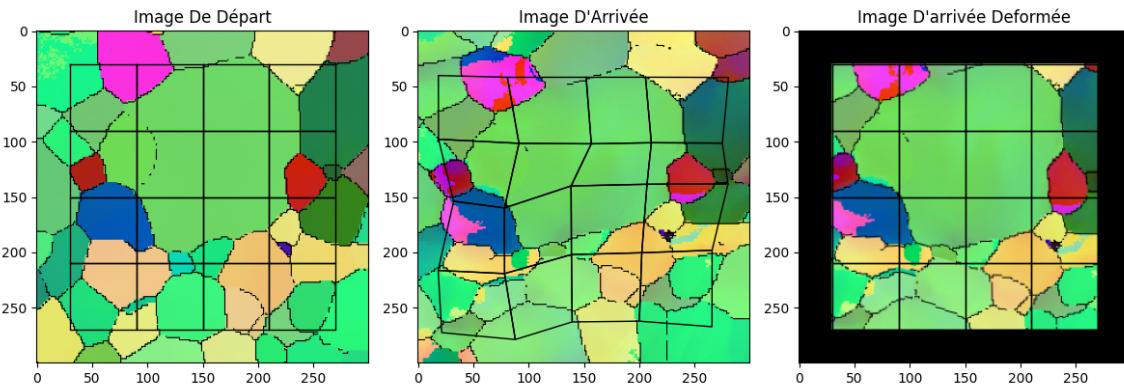


FIGURE 14 – Exemple d’application de la méthode

La figure (14) illustre l’application de la méthode, montrant comment l’image d’arrivée est déformée en utilisant l’interpolation bilinéaire et les grilles pour obtenir une nouvelle version déformée de l’image d’arrivée.

La grille de départ est placée de manière à montrer la région d’intérêt que nous souhaitons aligner. On peut clairement voir que pour les pixels contenus dans chaque quadrilatère de la grille d’arrivée, ils sont déplacés vers leur correspondant dans la grille uniforme de l’image d’arrivée déformée. Pour les pixels en dehors de la grille, ils vont avoir la couleur noire, ce qui forme la bordure noire dans l’image d’arrivée déformée puisque la transformation n’est définie que de la grille de départ vers la grille d’arrivée et la couleur noire est attribuée pour les pixels en dehors.

3.3 Déformation de la grille d’arrivée

Comme mentionné précédemment, la déformation de la grille d’arrivée est obtenue par un processus itératif, basé sur l’optimisation du critère d’alignement entre l’image de départ et l’image d’arrivée déformée. Cela implique la minimisation de ce critère, ce qui est équivalent à trouver une transformation géométrique efficace pour aligner les deux images. Cette déformation découle de la résolution d’un problème d’optimisation.

3.3.1 Problème d’optimisation

Un problème d’optimisation[12] est une tâche qui consiste à trouver la meilleure solution parmi un ensemble de solutions possibles. Dans notre contexte, le problème d’optimisation vise à déterminer la transformation géométrique optimale, définie par les deux grilles, qui minimise le critère d’alignement entre l’image de départ et l’image d’arrivée déformée. Cette transformation doit être capable d’ajuster la grille d’arrivée pour obtenir la meilleure correspondance possible entre les deux images après la transformation de l’image d’arrivée selon cette grille d’arrivée. Il est important de noter que, tout au long du processus d’optimisation, la grille de départ reste immuable, tandis que seule la grille d’arrivée subit des ajustements.

3.3.2 Descente de gradient

La descente de gradient[11] est l'une des méthodes couramment utilisées pour résoudre des problèmes d'optimisation. C'est un algorithme d'optimisation itératif qui ajuste progressivement ces paramètres dans notre contexte de la transformation géométrique, les paramètres sont les coordonnées des points composant la grille d'arrivée, pour minimiser le critère d'alignement. L'idée principale est de suivre un chemin dans l'espace des paramètres de la transformation qui conduit à une valeur minimale du critère.

Au cours de chaque itération de la descente de gradient, les paramètres (les coordonnées des noeuds de la grille d'arrivée) sont mis à jour en fonction des gradients du critère par rapport à ces paramètres. Cette mise à jour itérative se répète jusqu'à ce que la convergence soit atteinte, indiquant que la meilleure transformation géométrique a été trouvée.

Mathématiquement le problème se traduit comme suit :

$$\min_{w \in \mathbb{R}^{d \times 2}} L(w) \quad (2)$$

L'équation ci-dessus représente la minimisation de la fonction $L(w)$ le critère d'alignement par rapport à la variable w où w est le vecteur contenant les coordonées des points de la grille d'arrivée et d est le nombre des noeuds de la grille $\mathbb{R}^{d \times 2}$.

L'algorithme de descente de gradient est la méthode la plus classique en optimisation dérivable. Elle se base sur le principe élémentaire suivant, tiré de la condition d'optimalité suivante :

Pour tout point $w \in \mathbb{R}^{d \times 2}$:

1. Soit $\nabla L(w) = 0$, auquel cas w est potentiellement un minimum local (et global si L est convexe).
2. Soit $\nabla L(w) \neq 0$, et on peut alors montrer que la fonction L décroît localement dans la direction de l'opposé du gradient $-\nabla L(w)$.

La seconde propriété constitue l'essence même de l'algorithme de descente de gradient.

L'algorithme de descente de gradient est un processus itératif qui se base sur la formule suivante :

$$w \leftarrow w - \alpha \nabla L(w), \quad (3)$$

où $\alpha > 0$ est un paramètre appelé taille de pas ou longueur de pas.

En utilisant la règle (3) au sein d'un processus itératif, on peut construire un algorithme dont le but consiste à minimiser la fonction L : il s'agit de l'algorithme de descente de gradient.

Algorithm 1 Algorithme de descente de gradient

Choisir $w_0 \in \mathbb{R}^{d \times 2}$

Initialiser w_0 avec une valeur appropriée

while non convergence **do**

 Calculer le gradient $\nabla L(w_k)$

 Choisir une longueur de pas $\alpha_k > 0$

 Mettre à jour $w_{k+1} = w_k - \alpha_k \nabla L(w_k)$

end

Critère d'arrêt Dans la pratique, un algorithme est soumis à des contraintes de budget (en termes d'opérations arithmétiques, de temps d'exécution, d'itérations). Des critères d'arrêt de la méthode sont donc nécessaires afin de forcer l'algorithme à terminer si ces limites sont atteintes. Pour l'algorithme (1), il pourrait s'agir de terminer la méthode après avoir effectué k_{max} itérations, auquel cas $w_{k_{max}}$ représenterait la meilleure solution obtenue. Par ailleurs, pour mesurer la capacité d'un algorithme à converger vers une solution du problème, on introduit

généralement un critère d'arrêt basé sur les conditions d'optimalité. Par exemple, le critère suivant est très fréquemment utilisé pour les algorithmes basés sur le gradient : $\|\nabla L(w_k)\| < \epsilon$ où $\epsilon > 0$ représente une précision donnée, de sorte que ce critère est plus difficile à vérifier lorsque ϵ est très faible.

Initialisation La performance de l'algorithme peut être grandement améliorée lorsque l'initialisation est bien choisie. Dans notre contexte, le placement de la grille d'arrivée est crucial. Plus la grille d'arrivée est choisie de manière à couvrir la même zone (au niveau des grains) que la grille de départ, meilleur est le résultat d'alignement obtenu.

Choix de longueur de pas Le choix de la taille de pas est important pour la performance de l'algorithme. C'est le taux d'apprentissage (Learning Rate) qui est un hyperparamètre à régler en fonction du problème. Dans la plupart des problèmes d'optimisation, l'inclusion de cet hyperparamètre est essentielle car son impact sur la convergence de l'algorithme est significatif. Il contrôle la taille des pas que l'algorithme effectue dans la direction du gradient, ce qui peut influencer la vitesse de convergence et la stabilité de l'optimisation.

Le choix du taux d'apprentissage dans un algorithme d'optimisation comporte des avantages et des inconvénients selon sa valeur :

1. **Taux d'apprentissage très petit :**

- **Avantages** : Un taux d'apprentissage très faible permet des pas de mise à jour prudents, garantissant généralement la convergence.
- **Inconvénients** : Cependant, cela peut entraîner une convergence lente et nécessiter un grand nombre d'itérations pour atteindre une solution optimale.

2. **Taux d'apprentissage très élevé :**

- **Avantages** : Un taux d'apprentissage élevé accélère la convergence, permettant de trouver rapidement une solution optimale.
- **Inconvénients** : Cependant, il comporte des risques de divergence, d'oscillation, ou de saut par-dessus des optima locaux.

Dans notre problème d'alignement, nous avons opté pour un taux d'apprentissage faible. Cela permet de minimiser le risque de divergence dû à la non-convexité des quadrilatères de la grille d'arrivée au cours des itérations. Si un taux d'apprentissage élevé était utilisé, la déformation de la grille pourrait potentiellement entraîner la non-convexité de ces quadrilatères, compliquant ainsi le processus d'optimisation puisque l'efficacité de l'interpolation bilinéaire peut être compromis face à ce problème.

3.3.3 ADAM

Adam[7], (acronyme pour Adaptive Moment Estimation) est un algorithme d'optimisation stochastique qui combine des méthodes de descente de gradient et d'ajustement adaptatif des taux d'apprentissage.

Le fonctionnement d'Adam repose sur deux moments : le premier moment (moyenne mobile) et le deuxième moment (moyenne mobile des carrés des gradients). Ces moments sont calculés itérativement pour chaque paramètre. Adam ajuste également les taux d'apprentissage de manière adaptative pour chaque paramètre, ce qui permet de converger plus rapidement vers le minimum global du critère d'alignement.

Nous avons utilisé l'algorithme Adam pour optimiser le critère d'alignement.

Algorithm 2 Algorithme Adam

Entrées :

- Taux d'apprentissage lr : contrôle la taille du pas lors des mises à jour des paramètres.
- β_1, β_2 : hyperparamètres contrôlant les taux de décroissance exponentielle pour les premiers et seconds moments.
- Critère d'alignement L
- Paramètre w

Initialiser :

- $t \leftarrow 0$: compteur d'itération.
- $m_0 \leftarrow 0$: initialisation du premier moment.
- $v_0 \leftarrow 0$: initialisation du deuxième moment.

while *non convergence* **do**

- $t \leftarrow t + 1$: incrémentation du compteur d'itération.
- Calculer le gradient : $g_t \leftarrow \nabla L(w_{t-1})$.
- Mettre à jour les moments :
 - $m_t \leftarrow \beta_1 m_{t-1} + (1 - \beta_1) g_t$.
 - $v_t \leftarrow \beta_2 v_{t-1} + (1 - \beta_2) g_t^2$.
- Corriger les biais :
 - $\hat{m}_t \leftarrow \frac{m_t}{1 - \beta_1^t}$.
 - $\hat{v}_t \leftarrow \frac{v_t}{1 - \beta_2^t}$.
- Mettre à jour les paramètres : $w_t \leftarrow w_{t-1} - lr \times \frac{\hat{m}_t}{\sqrt{\hat{v}_t + \epsilon}}$.

end

3.4 Récapitulatif des phases du processus de la méthode proposée

1. Initialisation des grilles

Le premier pas de notre méthode d'alignement consiste à l'initialisation des grilles de départ et d'arrivée.

Positionnement précis : Nous superposons les grilles de départ et d'arrivée sur les images aux emplacements souhaités. Il est essentiel de choisir soigneusement ces emplacements, car ils doivent permettre une superposition optimale des deux images. L'un des principaux objectifs est de garantir que les grilles couvrent la même zone (au niveau des grains) que nous souhaitons aligner.

Remarque : Il est essentiel de noter que certaines conditions s'appliquent à l'initialisation des grilles. Tout d'abord, les deux grilles doivent avoir le même nombre de noeuds pour garantir une correspondance entre les quadrillatères. De plus, au début du processus, la méthode impose une initialisation uniforme des grilles.

2. Réglage des hyperparamètres

Ensuite, nous réglons les hyperparamètres pour obtenir un bon alignement et essayer de réduire autant que possible les risques de divergence. L'un des plus importants hyperparamètres est le taux d'apprentissage.

On peut préciser un nombre maximum d'époques pour le processus itératif. On peut également ajouter des conditions pour arrêter l'algorithme avant d'atteindre ce nombre d'époques si l'on satisfait une condition sur le critère d'alignement.

3. Entraînement

L'optimiseur Adam est utilisé pour minimiser le critère d'alignement que nous avons défini. L'entraînement se déroule sur un certain nombre d'époques.

À chaque itération, la méthode suit les étapes suivantes :

- (a) Au départ, nous avons deux grilles : une grille de départ et une grille d'arrivée que nous avons initialisé au début. Nous effectuons la correspondance de chaque quadrilatère de la grille de départ avec son homologue dans la grille d'arrivée en utilisant l'interpolation bilinéaire par quadrilatère, ce qui résulte en une transformation géométrique globale. Cette transformation est ensuite appliquée à l'image d'arrivée, créant ainsi une image d'arrivée déformée.
- (b) Une fois que l'image d'arrivée déformée est construite, le critère d'alignement est calculé entre cette dernière et l'image de départ.
- (c) À ce stade, l'optimiseur entre en jeu. Il calcule les gradients par rapport aux paramètres et met à jour les coordonnées des points de la grille d'arrivée. La grille de départ demeure fixe tout au long des époques.
- (d) Après avoir mis à jour la grille d'arrivée, une nouvelle itération commence avec cette nouvelle grille.

4. Convergence et visualisation

Une fois que l'algorithme a convergé, que ce soit en complétant le nombre d'époques prévu ou en s'arrêtant en fonction d'un critère prédéfini, nous procédons à la visualisation des résultats pour évaluer l'alignement. Au cours des itérations, nous sauvegardons toutes les images intermédiaires afin d'atteindre l'alignement optimal.

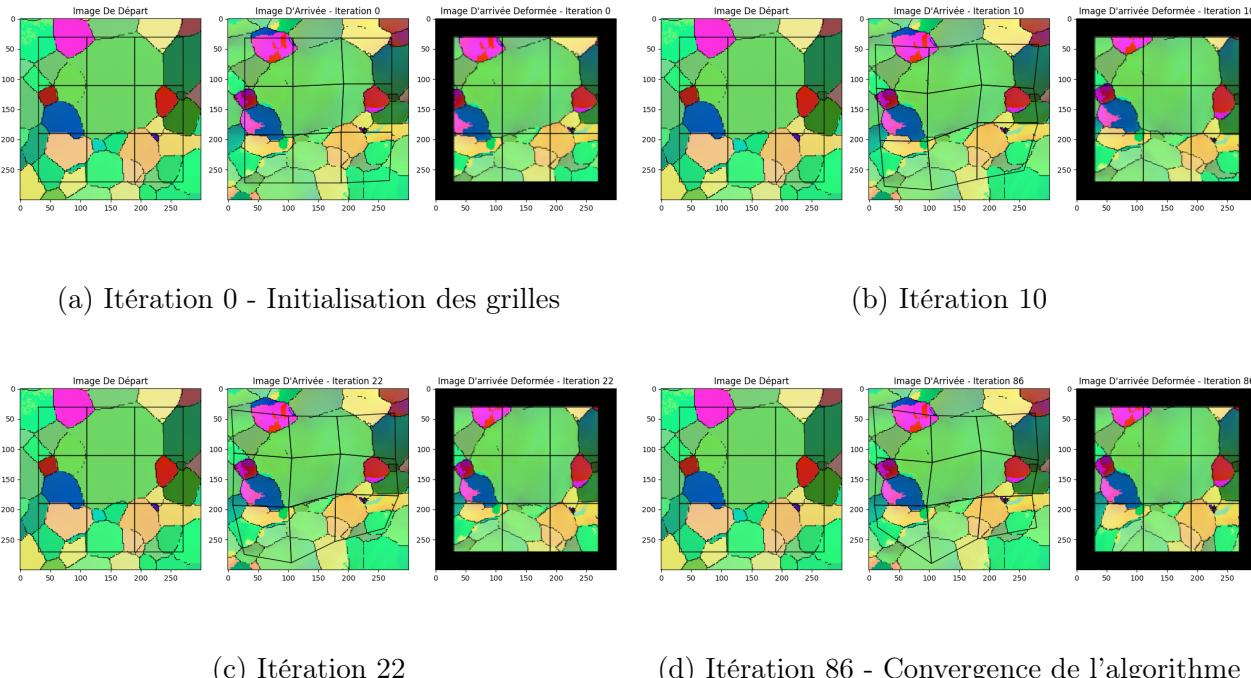


FIGURE 15 – Visualisation des itérations du modèle.

La figure (15) présente une visualisation du processus itératif. La figure (15)(a) montre l'initialisation des grilles à l'itération 0. À gauche, on trouve l'image de départ avec la grille, la zone de l'image à l'intérieur de la grille de départ que l'on souhaite aligner. Au milieu, on a l'image d'arrivée avec la grille d'arrivée. L'algorithme impose une initialisation uniforme des grilles. À droite, on a l'image d'arrivée déformée à l'itération zéro. Cette dernière sera identique à l'image d'arrivée puisque la grille est uniforme.

La figure (15)(b) représente la dixième itération de l'algorithme. La grille de départ n'a subi aucune déformation, c'est la grille d'arrivée qui a été déformée. Suite à la déformation de la grille d'arrivée, une nouvelle image d'arrivée déformée se produit. On peut observer que le gros grain en vert au milieu était d'une forme ovale dans l'image d'arrivée déformée à l'itération 0, et il devient d'une forme ronde à la dixième itération. En comparaison avec l'image de départ, on peut voir qu'elle s'approche. Dans les prochaines itérations, cette image d'arrivée déformée sera plus proche de l'image de départ.

La figure (15)(c) représente l'itération 22 de l'algorithme, et enfin, la figure (15)(d) représente l'itération 86, marquant la convergence de l'algorithme. Ces illustrations démontrent comment les grilles se déforment au fil des itérations pour déformer l'image d'arrivée afin qu'elle se superpose au mieux avec l'image de départ.

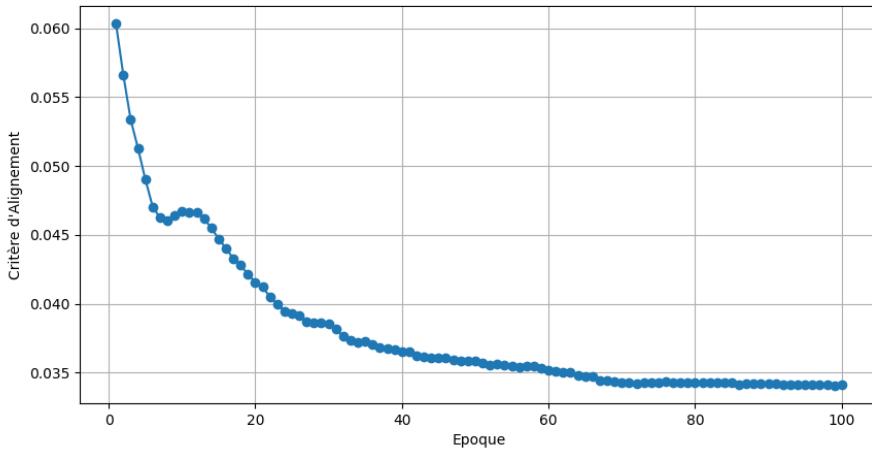


FIGURE 16 – Évolution du critère d'alignement en fonction des époques

La figure (16) illustre visuellement la variation des valeurs du **critère d'alignement** sur plusieurs époques au cours d'un processus d'entraînement.

Au fur et à mesure que l'on progresse de gauche à droite (représentant le passage des époques), l'axe des ordonnées affiche les valeurs du critère d'alignement, tandis que l'axe des abscisses présente les numéros d'époque correspondants.

Le graphique commence généralement avec une valeur élevée et diminue progressivement au fil des époques, indiquant que le modèle s'améliore. La trajectoire descendante de la ligne signifie la convergence du modèle vers un état où il minimise le critère d'alignement.

Cette représentation visuelle joue un rôle essentiel dans la surveillance et l'évaluation du processus d'entraînement. Elle nous aide à évaluer l'efficacité de la méthode.

3.5 Avantages de la méthode

- **Rapidité** : La méthode d'alignement proposée est relativement rapide par rapport aux méthodes existantes[6].
- **Nombre de paramètres limité** : La méthode implique un nombre restreint de paramètres à optimiser, ce qui facilite sa mise en œuvre et réduit les risques de sur-ajustement.
- **Expressivité de la transformation** : Les transformations basées sur les grilles sont suffisamment expressives pour modéliser la déformation résultant du laminage ou la traction uniaxiale, tout en restant paramétrées de manière contrôlable.
- **Faible besoin en données** : Contrairement à certaines méthodes basées sur l'apprentissage automatique, cette méthode ne nécessite pas un grand volume de données pour

fonctionner efficacement. Cela s'avère particulièrement avantageux dans des cas où les données expérimentales sont rares ou limitées.

3.6 Inconvénients et limitations

- **Mise à jour locale du paramètre :** La méthode repose sur l'utilisation du gradient pour déterminer la direction de déplacement. Cependant, le gradient ne tient compte que des variations locales, ce qui peut entraîner une convergence vers un optimum local et ne pas garantir la solution optimale globale.
- **Sensibilité à l'initialisation :** L'efficacité de la méthode dépend largement de l'initialisation des grilles. Des études approfondies sur l'impact de l'initialisation sont nécessaires pour mieux comprendre ses limitations et envisager des améliorations.
- **Perte de convexité :** La méthode ne garantit pas que les quadrilatères restent convexes lors de l'optimisation, ce qui peut poser problème pour l'interpolation bilinéaire.
- **Non-prise en compte des symétries cristallines :** Le critère d'alignement actuel ne tient pas compte des symétries des cristaux. Lors de la déformation d'un grain, les symétries peuvent entraîner des variations des angles d'Euler pour les pixels à l'intérieur du même grain, ce qui se traduit par des variations de couleurs au sein du même grain. Cela introduit un biais qui peut influencer négativement l'algorithme.

Cependant, ce problème peut être résolu en utilisant des images du matériau construites à partir des angles de désorientation. Les angles de désorientation mesurent l'écart angulaire entre les orientations cristallographiques de deux grains adjacents dans un matériau polycristallin. Les images basées sur les angles de désorientation représentent chaque grain avec une seule couleur pour tous ses pixels, éliminant ainsi les variations de couleurs dues aux symétries cristallines.

- **Sensibilité de la structure cristalline aux contraintes :** La méthode actuelle est qu'elle ne prend pas en compte la réaction des grains cristallins sous l'influence de contraintes importantes. Dans de telles conditions, les grains ont tendance à se diviser et de nouveaux cristaux peuvent apparaître. Cette évolution de la structure cristalline n'est pas prise en considération par l'algorithme actuel, ce qui peut conduire à des incohérences dans les résultats, en particulier lorsque l'on s'intéresse à des matériaux soumis à des contraintes importantes.

Mais cela est résolu si l'on travaille par paliers : on applique l'algorithme plusieurs fois avec des contraintes de plus en plus importantes.

3.7 Améliorations envisageables

- **Test d'initialisations multiples :** Une approche pourrait consister à tester différentes initialisations des grilles et à sélectionner celle qui obtient les meilleurs résultats en termes de critère d'alignement.
- **Initialisation intelligente :** L'utilisation de points de référence pour une initialisation plus intelligente pourrait améliorer les performances. Des techniques telles que SIFT[14] (Scale-Invariant Feature Transform) ou la correspondance de points triples[15] pourraient être explorées.
- **Maintien de la convexité :** Il est possible d'explorer des approches pour garantir la convexité des quadrilatères tout au long de l'optimisation, que ce soit par l'ajout d'une pénalité dans le critère ou en utilisant une solution projetée, c'est-à-dire en restreignant les modifications aux noeuds de manière à ce qu'ils demeurent dans l'espace des solutions admissibles qui préservent la convexité.

3.8 Perspectives futures

À long terme, l'objectif de ce travail préparatoire est de mettre en place un modèle d'apprentissage automatique qui pourrait prédire les déformations. Pour cela, il faudra obtenir le meilleur alignement possible.

Avant de mettre en œuvre un modèle d'apprentissage automatique, une étape importante doit être réalisée : **la préparation des données**.

Dans le contexte de la prédiction des déformations des grains, l'alignement assure que les caractéristiques correspondantes entre les images de départ et d'arrivée soient correctement superposées. Il est important que les images de départ et d'arrivée couvrent la même zone (les mêmes grains). Les grains présents dans une image qui ne sont pas présents dans l'autre image créent un biais. Ce sont des données aberrantes qui peuvent fausser les prédictions. Cela permet au modèle d'apprentissage automatique de mieux comprendre les relations entre les caractéristiques des images et les déformations correspondantes.

De plus, un bon alignement peut aider à augmenter la quantité de données utilisables pour l'entraînement du modèle. Par exemple, si on a plusieurs images de la même zone à différents niveaux de contrainte appliquée, un bon alignement peut nous permettre de les combiner pour créer un ensemble de données plus grand et plus diversifié. Ou bien, si on a une grande carte EBSD, ce qui est généralement le cas, on peut créer des tuiles en alignant de petites zones des cartes de départ et d'arrivée.

En utilisant la méthode actuelle, lorsque davantage de données seront disponibles, il sera envisageable de créer un modèle IA prédictif qui prédit les déformations des matériaux à partir d'une image d'un matériau au repos et d'une force appliquée en entrée et le champ de déformation en sortie, il faut un jeu de données contenant des images au repos et les champs de déformation permettant de construire l'image d'arrivée.

Or nous n'avons pas ces champ de déformation, on n'a que les images de départ, et les images d'arrivée.

la méthode d'alignement permet de produire les champs de déformation après avoir aligner les images, ces champs de déformation produits par la méthode nous permettent de concevoir un modele IA pour prédire les déformations .

4 Expériences et résultats

Dans cette section, nous présentons les résultats de notre méthode. Nous disposons de deux types de données : un matériau soumis au processus de laminage et un autre matériau soumis à la traction uniaxiale.

- Le laminage est un processus de fabrication qui implique la réduction de l'épaisseur d'un matériau en le faisant passer entre deux rouleaux. Il est couramment utilisé pour produire des feuilles de métal, des tôles, et d'autres produits plats.
- La traction uniaxiale est une traction suivant une seule direction, ce qui signifie que le matériau est étiré dans une seule direction à partir de ses extrémités.

Nous allons tester l'algorithme sur ces deux types de données et analyser sa performance.

4.1 Laminage

Dans cette section, nous présenterons les résultats de nos expériences liées au processus de laminage. Nous avons soumis **l'acier** à ce processus et testerons notre algorithme sur ces données. Nous analyserons l'influence de plusieurs paramètres, notamment la taille de la grille, l'initialisation, et l'impact du choix du taux d'apprentissage.

4.1.1 Position initiale de la grille d'arrivée

Pendant ces tests, la grille de départ est fixe, sa position indiquant la zone d'intérêt pour l'alignement, tandis que nous avons varié l'emplacement de la grille d'arrivée à chaque test pour évaluer les résultats. Les résultats de ces tests sont illustrés dans les deux figures ci-dessous.

Pour les tests suivants, nous avons maintenu la grille de départ au même emplacement pour les trois tests, avec un taux d'apprentissage égal à 8.

- Pour le premier test, nous nous concentrerons sur la région indiquée par la grille de départ, que nous cherchons à aligner. Cette région sera la même pour les autres tests, où nous ne changerons que l'emplacement de la grille d'arrivée.
- Pour le deuxième test, l'emplacement de la grille d'arrivée est le même que dans le premier test, mais il est décalé de 150 unités le long de l'axe des x.
- Pour le troisième test, l'emplacement de la grille d'arrivée est le même que dans le premier test, mais il est décalé de 110 unités le long de l'axe des x.

Numéro de Test	Taille de la zone à aligner	Critère d'alignement	Temps d'exécution (s)
1	1449 x 1449, grille(3x3)	0.05467	4200
2	1449 x 1449, grille(3x3)	0.05015	2340
3	1449 x 1449, grille(3x3)	0.04536	1980

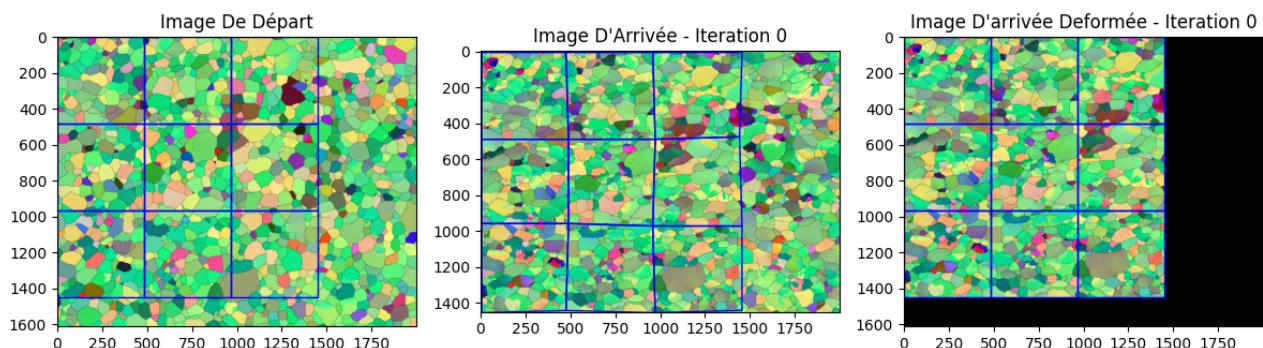


FIGURE 17 – Initialisation des grilles - Test 1.

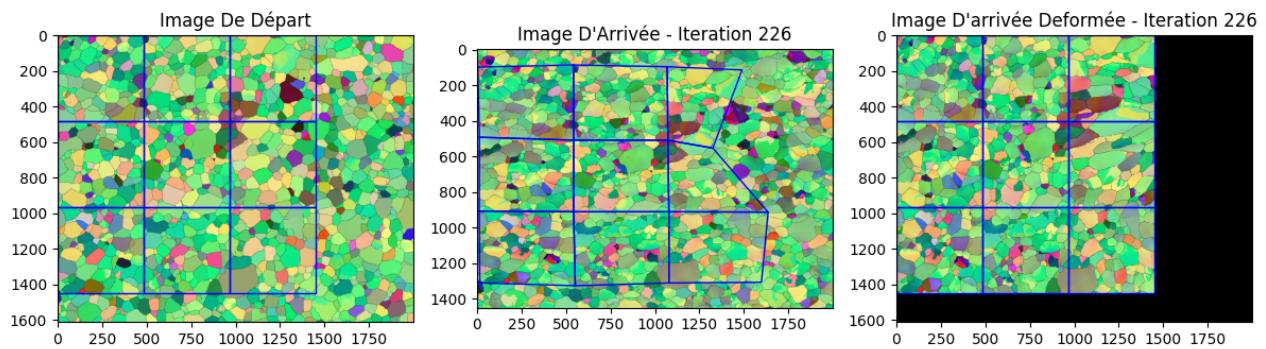


FIGURE 18 – Résultats de la convergence - Test 1.

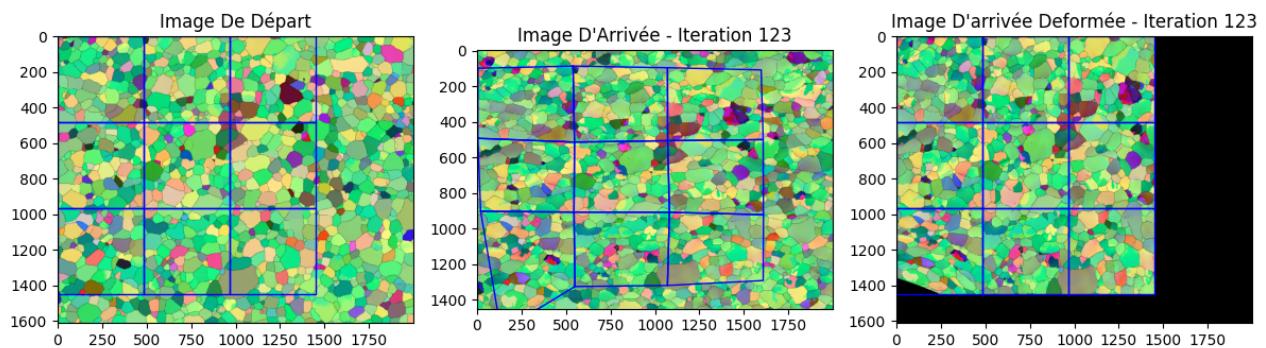


FIGURE 19 – Résultats de la convergence - Test 2.

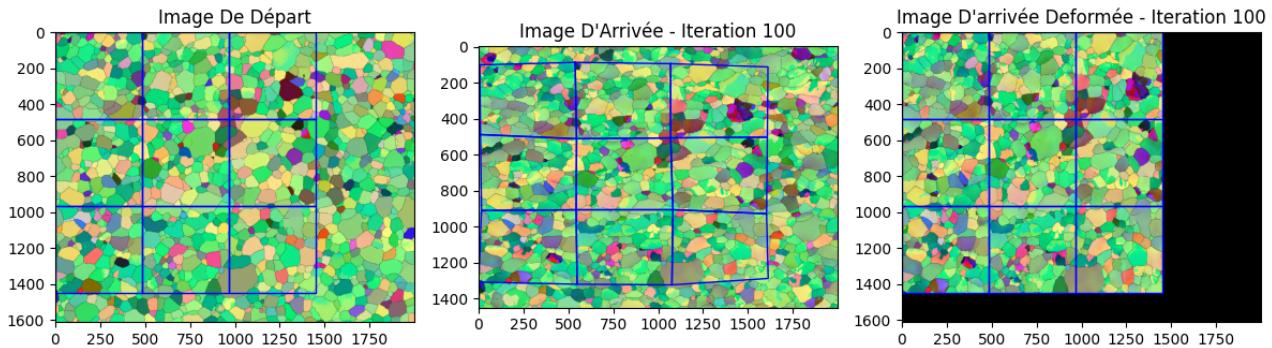


FIGURE 20 – Résultats de la convergence - Test 3.

Interprétation Pour le premier test, des problèmes d’alignement sont survenus pour les deux quadrilatères en haut à droite. L’initialisation que nous avons effectuée a introduit des grains à droite dans la grille de départ qui n’étaient pas présents dans la grille d’arrivée. C’est pourquoi nous avons tenté de translater la grille de 150 suivant l’axe des x lors du deuxième test. Cependant, des problèmes sont apparus pour le quadrilatère en bas à gauche. Pour le troisième test, nous avons effectué une translation légèrement en arrière de -40.

- La méthode est sensible à l’initialisation. Plus les deux grilles contiennent les mêmes grains à l’intérieur, meilleur est l’alignement et plus rapidement l’algorithme converge.

- On constate que la méthode fonctionne bien pour le troisième test, et la transformation géométrique a correctement modélisé la déformation du matériau suite au laminage. On observe clairement que la grille est étirée aux deux extrémités et aplatie.

Maintenant, nous allons effectuer un quatrième test en conservant les mêmes emplacements des grilles sur les deux images que le troisième test, mais en modifiant la taille de la grille de 4x4.

Numéro de Test	Taille de la zone à aligner	Critère d’alignement	Temps d’exécution (s)
4	1449 x 1449, grille(4x4)	0.04732	1020

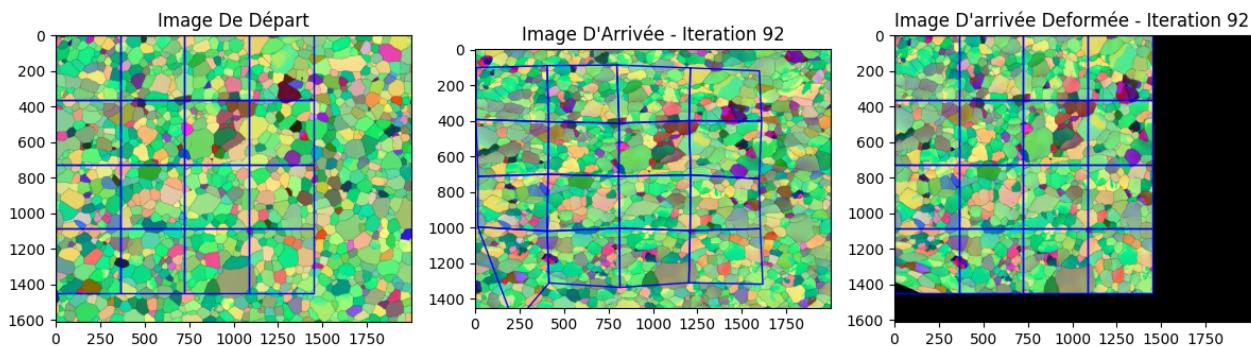
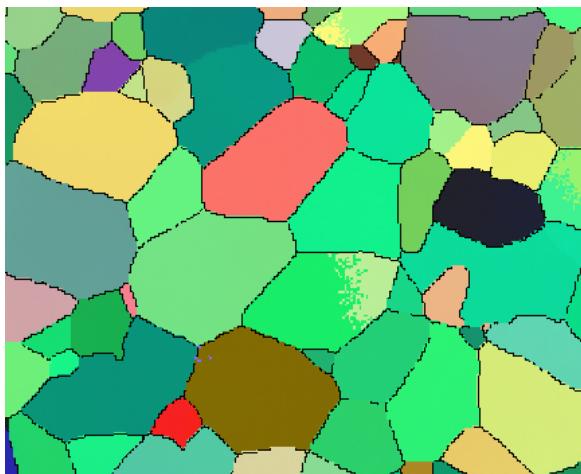
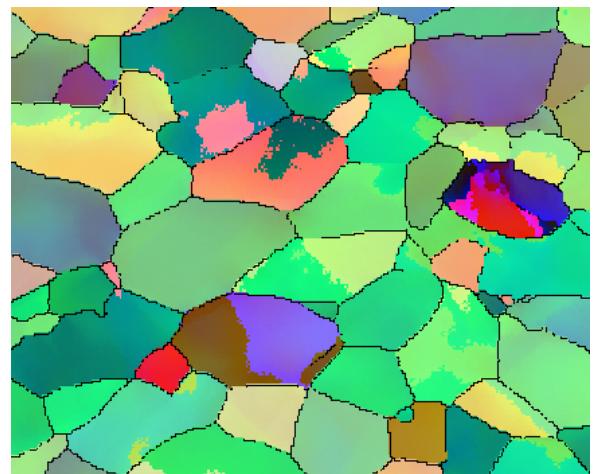


FIGURE 21 – Résultats de la convergence - Test 4.

Pour le quatrième test, nous rencontrons le même problème que dans le second test, particulièrement au niveau du quadrilatère en bas à gauche. En analysant cette zone de manière individuelle, on observe la présence de plusieurs grains de couleurs différentes dans l'image d'arrivée figure (22-b).



(a) Image de départ



(b) Image d'arrivée

FIGURE 22 – Le quadrilatère qui n'a pas été bien aligné pour les tests 2 et 4

Dans la figure (22), on peut observer que certains grains de l'image de départ étaient d'une seule couleur, mais après le laminage, d'autres couleurs sont apparues à l'intérieur. Cela signifie que l'orientation au sein du grain a changé, mais ces cristaux qui ont changé d'orientation restent liés au même grain en raison de la symétrie des cristaux. Il ne s'agit pas d'une fragmentation du grain.

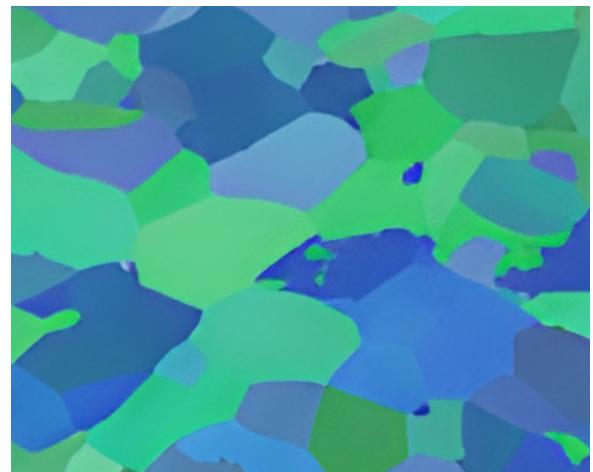
Ce changement de couleurs peut introduire un biais dans l'algorithme et conduire à des résultats incorrects. Pour remédier à cela, au lieu d'utiliser les angles d'Euler comme valeurs RGB, il est possible d'utiliser les angles de désorientation. Les angles de désorientation sont des mesures qui quantifient le degré de désalignement ou de rotation entre les cristaux adjacents dans un matériau polycristallin.

L'avantage de cette approche est que la variation de couleur au sein d'un grain est moins apparente. De plus, nous tentons d'utiliser des images sans les frontières des grains. Ces frontières peuvent introduire un biais dans l'algorithme. Par ailleurs, la réduction de la taille des images permet à l'algorithme de s'exécuter plus rapidement.

Il est à noter que les frontières noires dans les images sont générées par les logiciels de traitement des cartes EBSD, qui utilisent des algorithmes pour détecter les grains. Nous avons utilisé ATEX[15] pour traiter les images.



(a) Image de départ



(b) Image d'arrivée

FIGURE 23 – Le quadrilatère qui n'a pas été bien aligné pour les tests 2 et 4 avec les angles de désorientation et pas les angles d'Euler

Maintenant, nous allons refaire le test 2 avec ces images. La taille des images a été réduite.

Numéro de Test	Taille de la zone à aligner	Critère d'alignement	Temps d'exécution (s)
5	360 x 360, grille(3x3)	0.00781	61

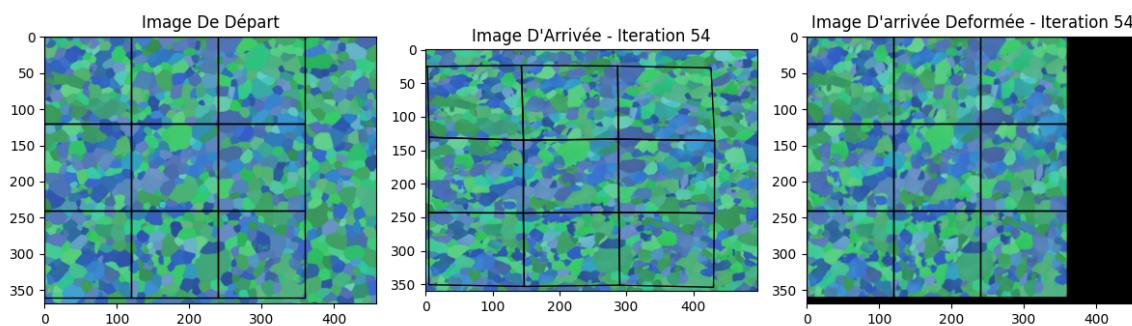


FIGURE 24 – Résultats de la convergence - Test 5.

Pour les prochains tests, nous allons utiliser ces images.

4.1.2 Variation de la taille de grille

Maintenant, pour la même région que celle observée dans les tests précédents, nous allons varier la taille de la grille. En ce qui concerne le taux d'apprentissage, nous allons le fixer à la valeur 3.

Numéro de Test	Taille de la zone à aligner	Critère d'alignement	Temps d'exécution (s)
6	360 x 360, grille(4x4)	0.00753	126
7	360 x 360, grille(5x5)	0.01275	233
8	360 x 360, grille(6x6)	0.01066	294
9	360 x 360, grille(7x7)	0.01077	179
10	360 x 360, grille(8x8)	0.01138	194

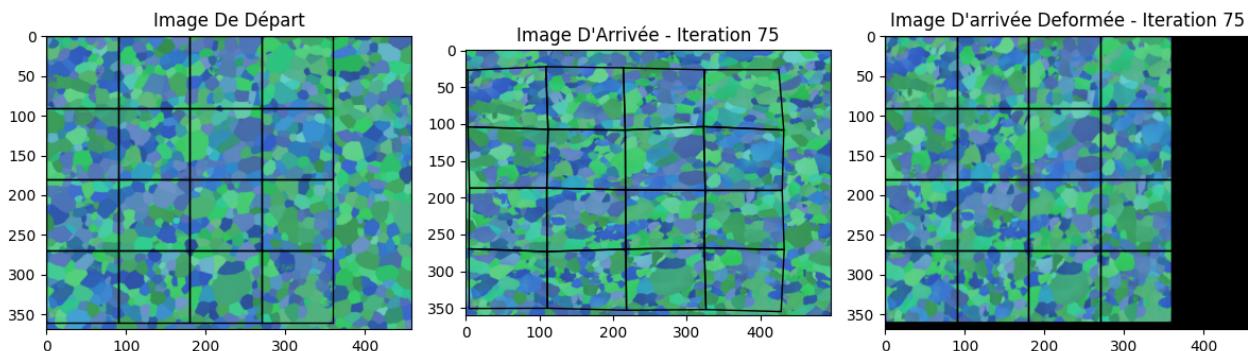


FIGURE 25 – Résultats de la convergence - Test 6.

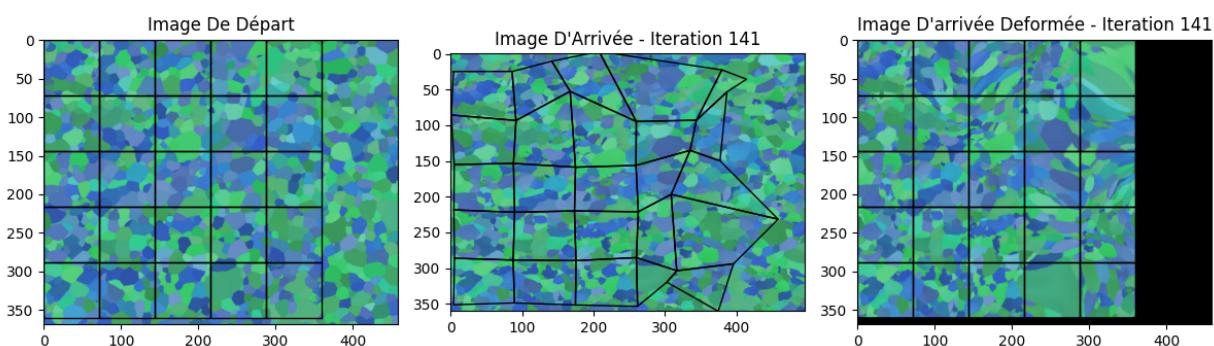


FIGURE 26 – Résultats de la convergence - Test 7.

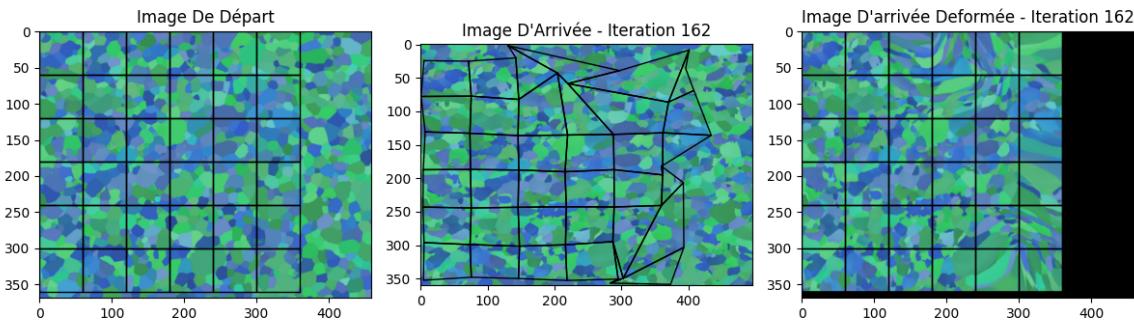


FIGURE 27 – Résultats de la convergence - Test 8.

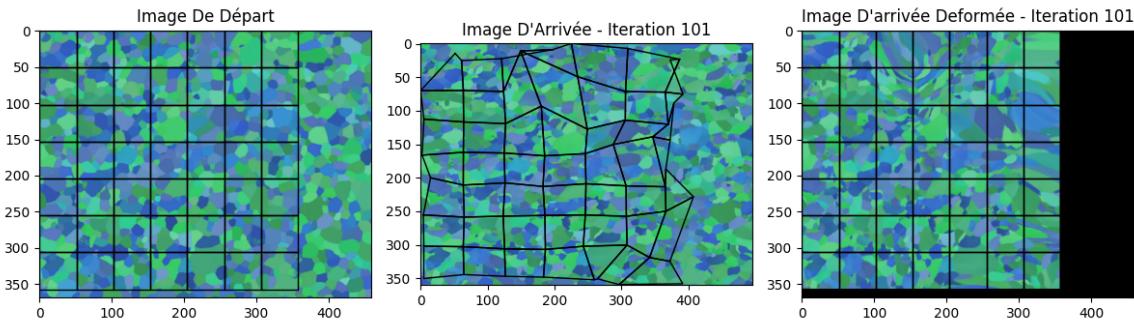


FIGURE 28 – Résultats de la convergence - Test 9.

Interprétation En augmentant la taille de la grille, il existe des risques de rencontrer des problèmes de non-convexité des quadrilatères déformés de la grille d'arrivée, surtout avec des taux d'apprentissage élevés. Une fois qu'on atteint un quadrilatère non-convexe, l'interpolation bilinéaire cesse de fonctionner correctement, et l'algorithme devient incapable de corriger ce problème.

Une solution pour résoudre ce problème est d'utiliser des taux d'apprentissage faibles et des initialisations très proches de la solution, de manière à ce que les mêmes grains contenus dans un quadrilatère de la grille de départ soient également présents dans son correspondant dans la grille d'arrivée. Cependant, cette approche peut parfois bloquer l'algorithme à un optimum local et ne pas parvenir à atteindre un alignement optimal.

Nous présentons ci-dessous trois tests illustratifs pour la même région, avec une grille de taille 5x5 :

Numéro de Test	Taux d'apprentissage	Critère d'alignement	Temps d'exécution (s)
11	1	0.01273	198
12	1.25	0.01201	221
13	1.25	0.01071	222

- Test 11 : Nous continuons à nous intéresser à la même zone de l'image de départ et testons avec une grille de taille 5x5, en fixant un taux d'apprentissage de 1. Cependant, l'algorithme atteint un optimum local et stagne.
- Test 12 : Toujours avec la grille de 5x5 sur la même zone, mais avec un taux d'apprentissage égal à 1,25, l'alignement s'améliore. L'algorithme a réussi à dépasser le minimum local rencontré dans le test 11. On peut encore améliorer ce résultat en déplaçant peut-être l'emplacement de la grille d'arrivée, ce que nous faisons dans le Test 13.
- Test 13 : Pour ce test, la grille d'arrivée à l'initialisation est la même que celle du test 12, mais elle est déplacée de 10 suivant l'axe des x.

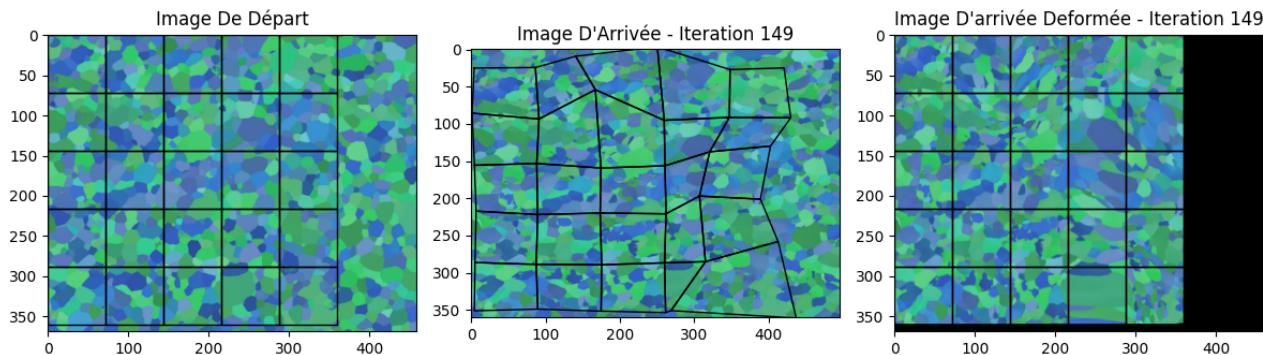


FIGURE 29 – Résultats de la convergence - Test 11.

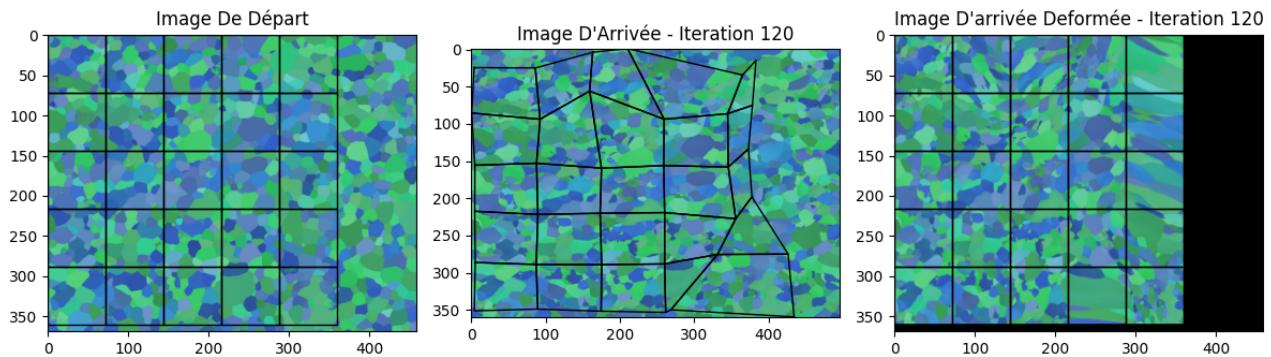


FIGURE 30 – Résultats de la convergence - Test 12.

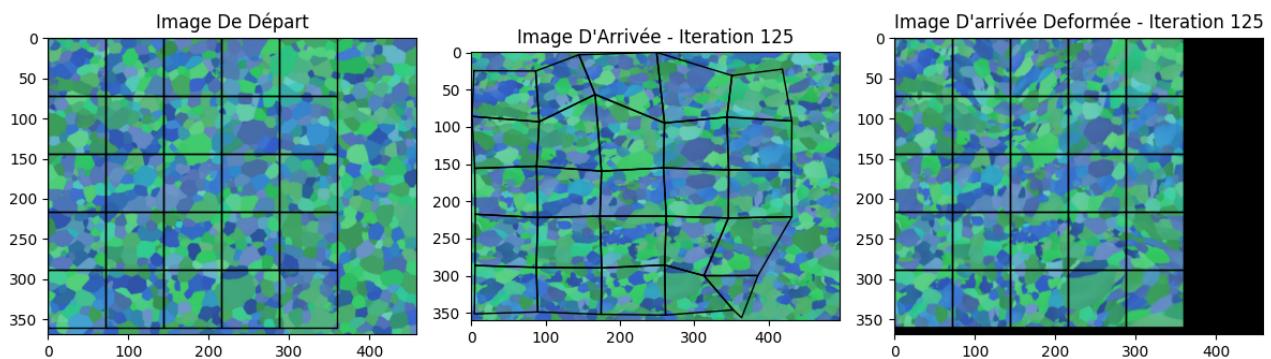


FIGURE 31 – Résultats de la convergence - Test 13.

4.2 Traction uniaxiale

Dans cette section, nous présenterons les résultats de nos expériences liées au processus de traction uniaxiale.

Description des Données En ce qui concerne ces données, nous avons utilisé l'inconel 625, un alliage très résistant aux hautes températures composé de nickel, de chrome et de fer. Il est largement utilisé dans l'industrie aéronautique.

Pour ce matériau, nous disposons d'images microscopiques à l'état initial, ainsi que d'images après avoir subi une traction uniaxiale avec des forces de 0.18 Newton, 0.28 Newton, 0.45 Newton, 1.3 Newton, 1.8 Newton, 2.8 Newton, 4 Newton, et 5.7 Newton.

Pour les tests effectués dans cette section, la représentation avec les angles de désorientation n'était pas claire, ce qui aurait compliqué la visualisation des grains. Par conséquent, nous allons procéder directement avec la présentation utilisant les angles d'Euler.

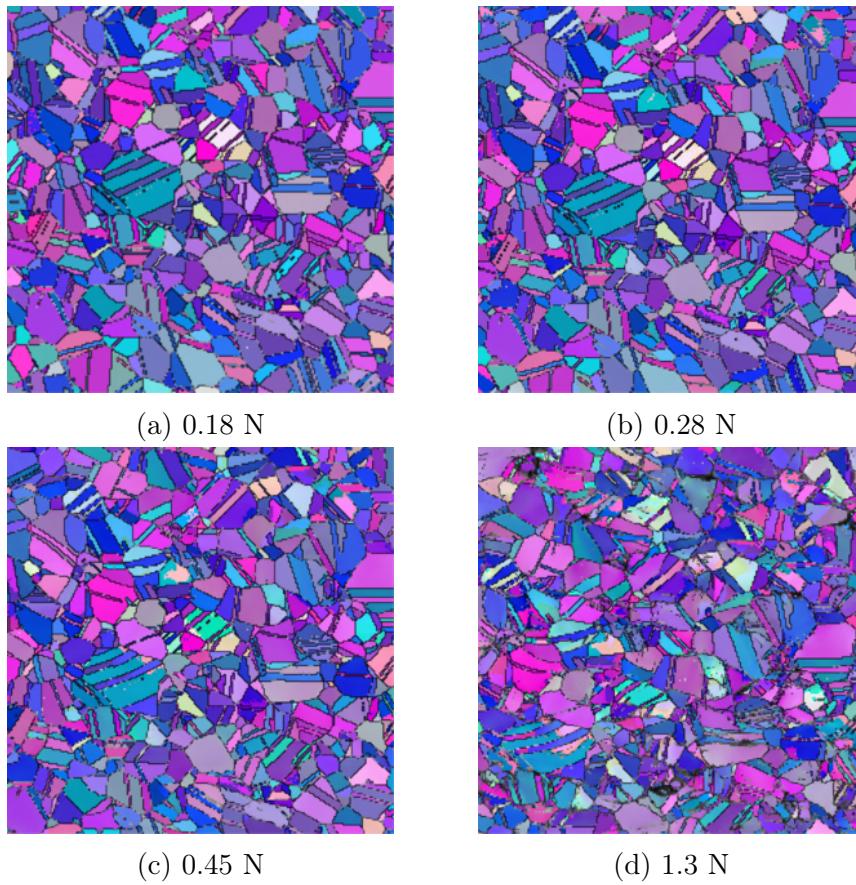


FIGURE 32 – Essais de traction uniaxiale sur inconel 625

4.2.1 Variation de la taille de grille

Dans cette sous-section, nous allons évaluer l'algorithme en utilisant les deux premières images de l'Inconel soumis à une traction uniaxiale avec des forces de 0.18 N et 0.28 N. Nous tenterons d'aligner une portion de l'image totale en faisant varier la taille de la grille. Le taux d'apprentissage a été fixé à 1 pour l'ensemble des tests.

Numéro de test	Taille de la zone à aligner	Critère d'alignement	Temps d'exécution (s)
14	200 x 200, grille(3x3)	0.01424	126
15	200 x 200, grille(4x4)	0.01419	126
16	200 x 200, grille(10x10)	0.01582	294

Interprétation En augmentant la taille de la grille, l'alignement s'améliore, rendant l'algorithme plus efficace pour capturer des déformations très fines, puisque la valeur du critère d'alignement diminue. Cependant, si la taille de la grille est augmentée de manière excessive, l'algorithme risque de surajuster, ne convergeant pas vers la solution appropriée et restant bloqué.

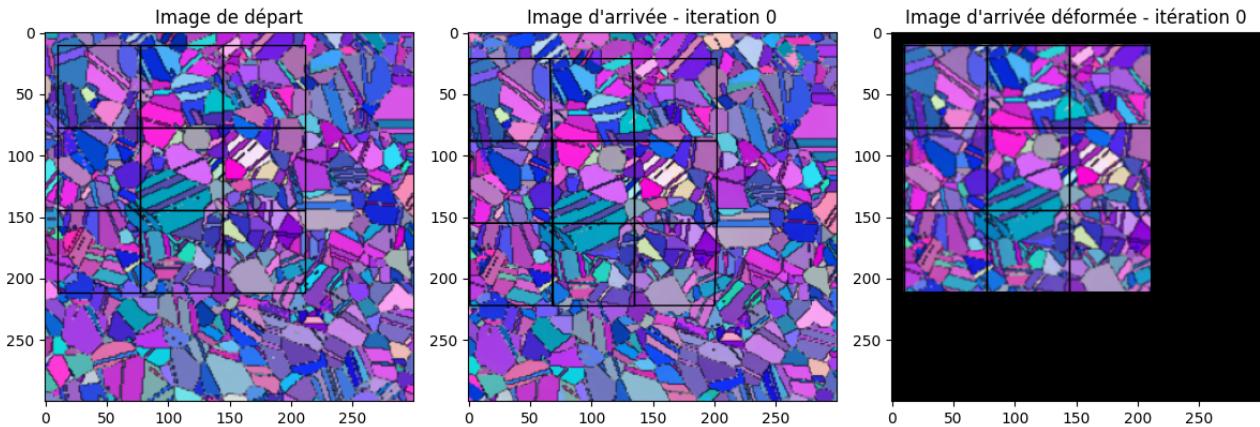


FIGURE 33 – Initialisation des grilles - Test 14.

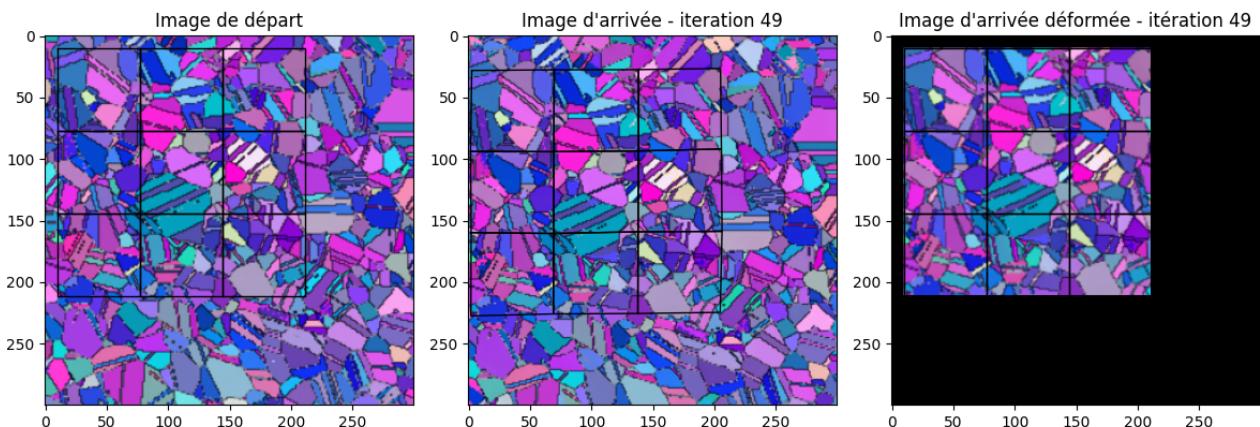


FIGURE 34 – Résultats de la convergence - Test 14.

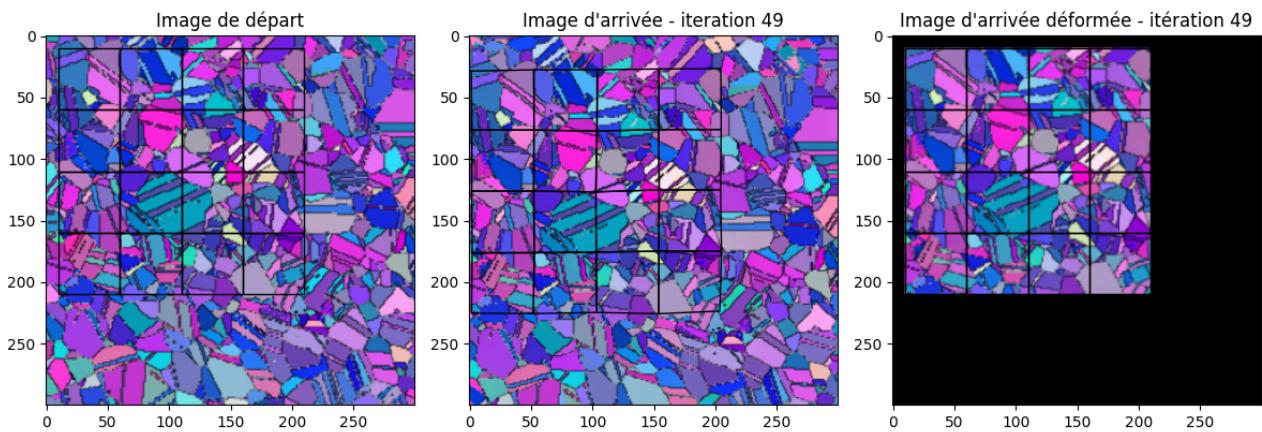


FIGURE 35 – Résultats de la convergence - Test 15.

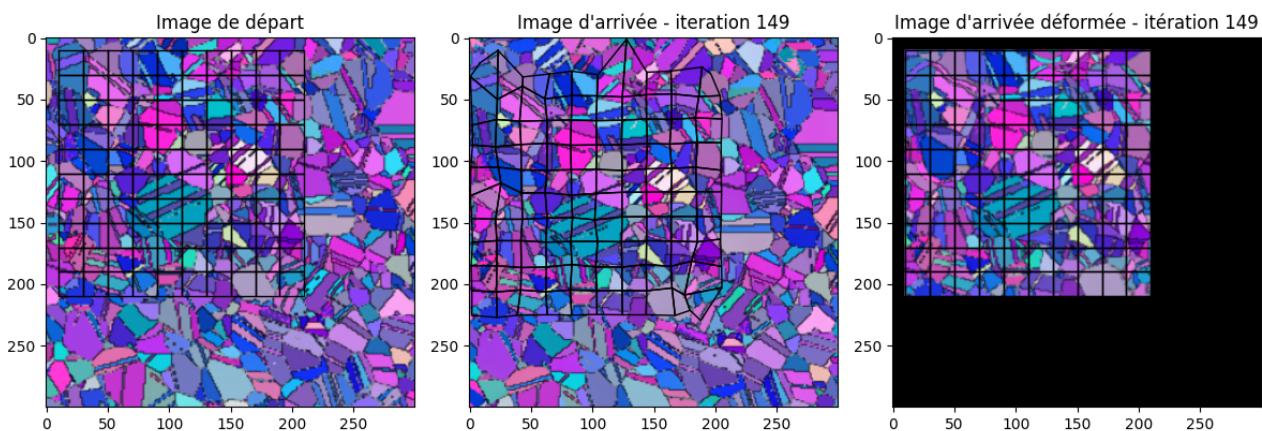


FIGURE 36 – Résultats de la convergence - Test 16.

4.2.2 Alignement de la première image avec les suivantes

Dans cette sous-section, nous allons tester l'algorithme pour voir s'il parvient à aligner des images présentant différents niveaux de déformation. Trois tests ont été réalisés pour aligner la première image avec les autres. Le taux d'apprentissage a été fixé à 2 pour ces tests, et la taille de la grille est fixée à 3x3. Il est à noter que, pour ces tests, j'ai essayé plusieurs initialisations des grilles et j'ai conservé la meilleure.

- **Test 17** : Alignement de la première image avec la deuxième.
- **Test 18** : Alignement de la première image avec la troisième.
- **Test 19** : Alignement de la première image avec la quatrième.

Numéro de test	Taille de la zone à aligner	Critère d'alignement	Temps d'exécution (s)
17	200 x 200	0.0135	180
18	200 x 200	0.0232	126
19	200 x 200	0.0414	126

Interprétation L'algorithme a correctement aligné les images pour les trois tests. En augmentant la force de la traction uniaxiale, la grille d'arrivée s'étire encore davantage, démontrant ainsi que la transformation appliquée est expressive pour illustrer la déformation du matériau lors de la traction uniaxiale. De plus, l'augmentation de la valeur du critère d'alignement indique une dégradation de la qualité de l'alignement en fonction de la force de contrainte. L'algorithme fonctionne de manière plus efficace pour des faibles déformations.

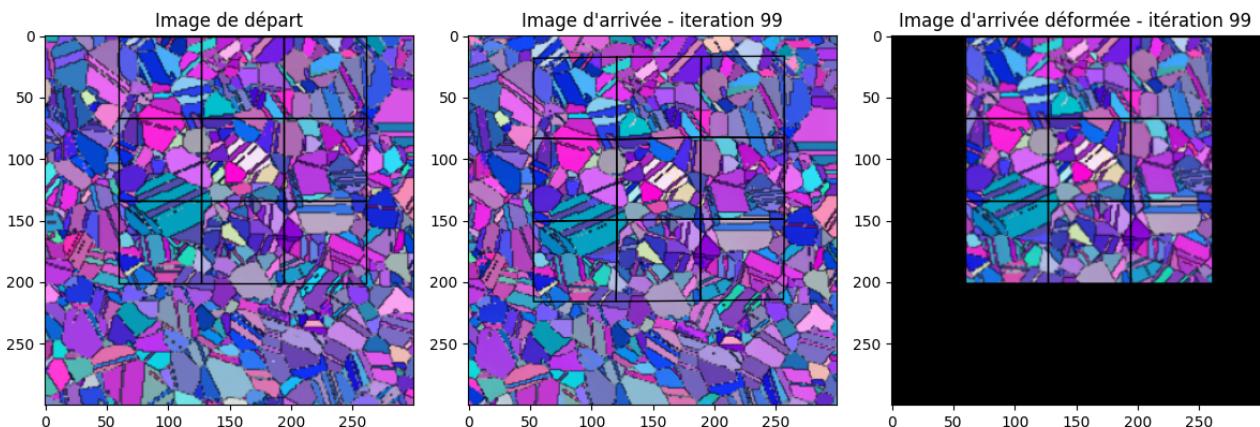


FIGURE 37 – Résultats de la convergence - Test 17.

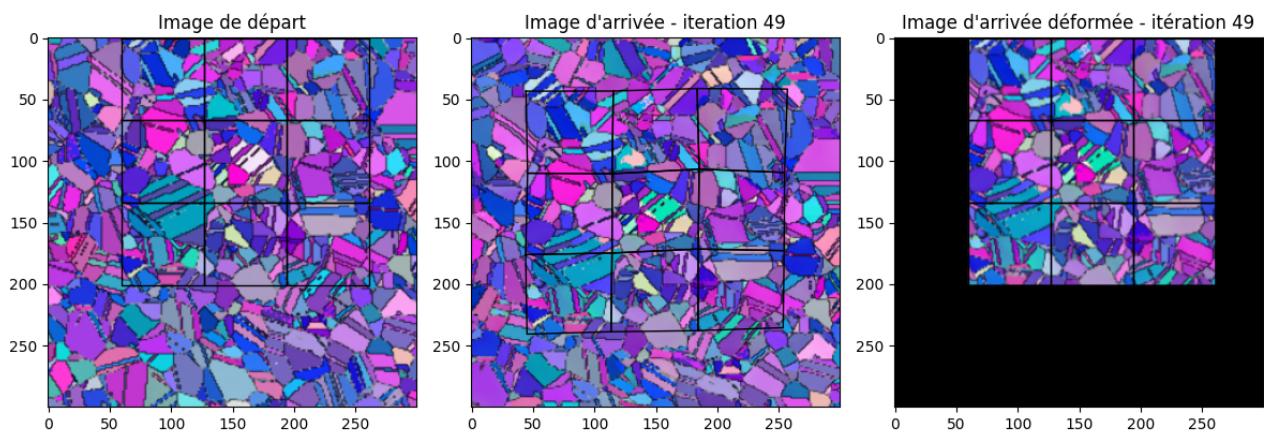


FIGURE 38 – Résultats de la convergence - Test 18.

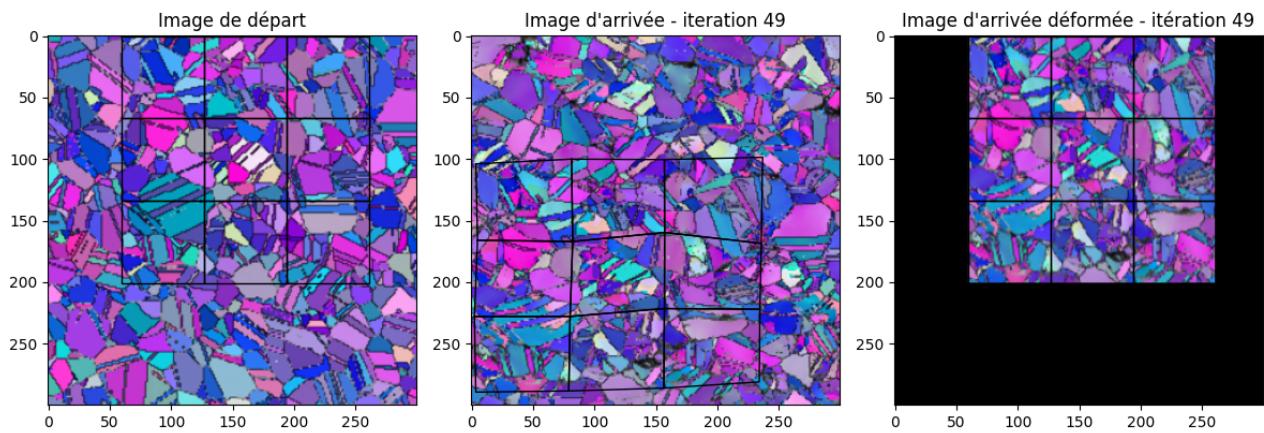


FIGURE 39 – Résultats de la convergence - Test 19.

Après avoir aligné les images pour ces derniers tests, on peut calculer les vecteurs de déplacement.

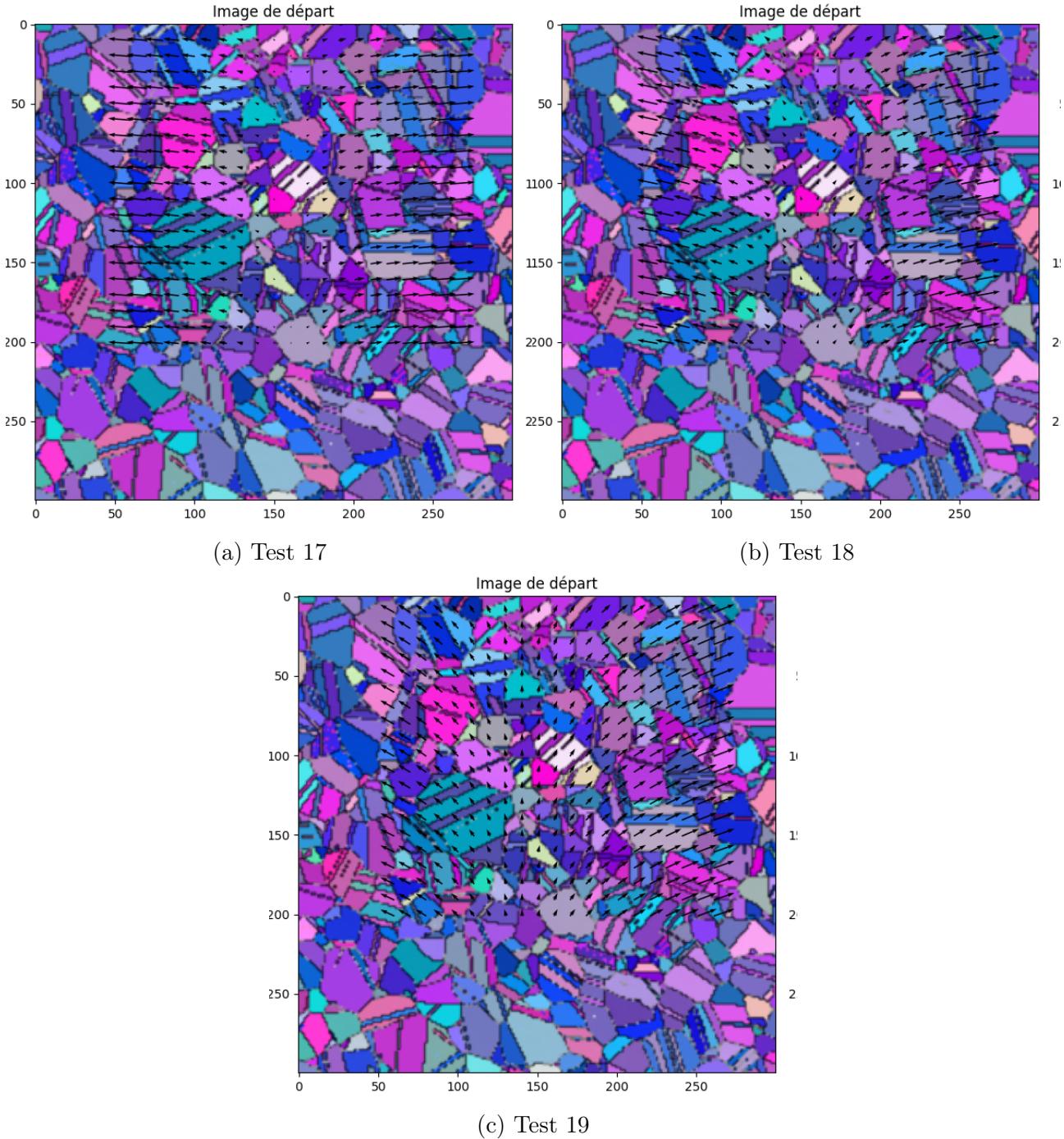


FIGURE 40 – Vecteurs de déplacement.

Les vecteurs de déplacement sont interprétés comme des indicateurs de la manière dont le matériau réagit à cette force. Lorsqu'on applique une force de traction uniaxiale aux deux extrémités d'un matériau, les extrémités du matériau ont tendance à subir un allongement significatif. En revanche, la partie centrale du matériau peut sembler rester relativement fixe ou subir un allongement moins prononcé. Cela est observable par les flèches des extrémités, qui sont plus étendues que celles de la partie centrale.

4.3 Conclusion

Nous avons évalué notre algorithme sur deux types de matériaux, à savoir l'acier et l'Inconel 625, et avons obtenu de bons résultats en termes d'alignement d'images, surtout en utilisant une grille de taille 3x3. En augmentant la taille de la grille, nous sommes capables de capturer des déformations très fines, mais il y a un risque de surajustement et quelques problèmes liés à la non-convexité des quadrilatères qui peuvent bloquer l'algorithme, rendant la transformation géométrique inefficace.

La transformation appliquée a efficacement capturé la déformation des matériaux soumis à des processus tels que le laminage ou la traction uniaxiale. Cependant, il est essentiel de souligner que notre méthode présente une forte dépendance à l'initialisation.

Après avoir aligné les deux images d'un matériau déformé, nous sommes en mesure de calculer les vecteurs de déplacement pour l'ensemble des points du matériau. Cette caractéristique se distingue de la méthode employée en science des matériaux (DIC Locale avec l'utilisation de mouchetis), les mouchetis[7] sont des taches noires ou blanches placées aléatoirement sur le matériau utilisés comme repères visuels lors de l'analyse de déformation par corrélation d'images numériques (DIC).

Cependant, un inconvénient majeur de la méthode DIC Locale utilisant les mouchetis réside dans la recherche d'une bonne configuration de mouchetis, ce qui peut parfois représenter un défi considérable. Et de ne pas pouvoir calculer les vecteurs de déplacement pour tous les points du matériau. En revanche, l'avantage principal de notre méthode réside dans sa rapidité à calculer les vecteurs de déplacement après l'alignement des deux images, couvrant ainsi tous les points du matériau. De plus, le nombre limité de paramètres facilite sa mise en œuvre.

Références

- [1] Documentation of EasySpin. *Rotations and Euler angles* an open-source MATLAB toolbox for simulating and fitting a wide range of Electron Paramagnetic Resonance (EPR) spectra. Disponible en ligne : <https://easyspin.org/documentation/eulerangles.html>.
- [2] Yang, Jin. *Fast Adaptive Augmented Lagrangian Digital Image Correlation 2019*. Disponible en ligne : https://www.researchgate.net/publication/331579806_Fast_Adaptive_Augmented_Lagrangian_Digital_Image_Correlation.
- [3] Masayuki Kamaya. *Assessment of local deformation using EBSD : Quantification of accuracy of measurement and definition of local gradient*. Disponible en ligne : <https://www.sciencedirect.com/science/article/abs/pii/S0304399111000726>.
- [4] *Corrélation d'images - Wikipedia*. Disponible en ligne : https://fr.wikipedia.org/wiki/Corrélation_d'images.
- [5] Christian Wachinger, Matthew Toews, Georg Langs, William Wells, Polina Golland. *La segmentation d'images médicales par la méthode du transfert de points d'intérêt*. Disponible en ligne : <https://substance.etsmtl.ca/segmentation-dimages-medicales-methode-du-transfert-de-points-dinteret>.
- [6] Hamid Reza Boveiri, Raouf Khayami, Reza Javidan, Ali Reza MehdiZadeh. *Medical Image Registration Using Deep Neural Networks : A Comprehensive Review*. Disponible en ligne : <https://arxiv.org/pdf/2002.03401.pdf>.
- [6] Dr. h.c. Dierk Raabe. *Digital Image Correlation (DIC) for local strain measurement in polycrystalline and inhomogeneous materials*. Disponible en ligne : <https://www.dierk-raabe.com/digital-image-correlation-dic/>.
- [7] Floriane SOULAS. *Comment réaliser un mouchetis adapté à la corrélation d'images numériques ?*. Disponible en ligne : <https://eikosim.com/articles-techniques/mouchetis-correlation-images-numeriques/>.
- [8] Sebastian Xhonneux. *Perception de l'optimisation en mathématiques et en économie au fil des siècles et l'enseignement du théorème de Lagrange*. Disponible en ligne : <https://archive.wikiwix.com/cache/index2.php?url=http%3A%2F%2Fwww.apmep.asso.fr%2FIMG%2Fpdf%2FLu-33-Xhonneux-LaRochelle.pdf%2Findex.html#federation=archive.wikiwix.com&tab=url>.
- [9] Max Mignotte. *TRAITEMENT D'IMAGES CORRECTION GÉOMÉTRIQUE ET TOMOGRAPHIE*. Disponible en ligne : https://www.iro.umontreal.ca/~mignotte/IFT6150/Chapitre7_ITF6150.pdf.
- [10] Rafael C. Gonzalez et Richard E. Woods *Digital Image Processing*, Pearson Prentice Hall, 2008, « *Image sampling and Quantization* », p. 66. . Disponible en ligne : <https://dl.ebooksworld.ir/motoman/Digital.Image.Processing.3rd.Edition.www.EBooksWorld.ir.pdf>.
- [11] Clément W. Royer *Optimisation pour l'apprentissage automatique p. 21*. Notes de cours - Master Intelligence Artificielle et Big Data Université Paris Dauphine-PSL
Disponible en ligne : <https://www.lamsade.dauphine.fr/~croyer/ensdocs/TUN/PolyTUN.pdf>.
- [12] Diederik P. Kingma and Jimmy Ba. *Adam : A method for stochastic optimization*. In International Conference on Learning Representations (ICLR), 2015. Disponible en ligne : <https://arxiv.org/abs/1412.6980>.
- [13] Jeremy Howard, Sylvain Gugger. *Deep Learning for Coders with fastai and PyTorch*. O'Reilly, 2020. ISBN 978-1492045526. Lire en ligne : <https://archive.org/details/DeepLearningForCodersWithFastAIAndPyTorch>.

wikiwix.com/cache/index2.php?url=https%3A%2F%2Fpython.engineering%2Fdeep-learning-for-coders-with-fastai-and-pytorch%2F#federation=archive.wikiwix.com&tab=url.

- [14] *Scale-invariant feature transform* - Wikipedia. Disponible en ligne : https://fr.wikipedia.org/wiki/Scale-invariant_feature_transform.
- [15] B. Beausir, J.-J. Fundenberger, *Analysis Tools for Electron and X-ray diffraction, ATEX - software*. Université de Lorraine - Metz, 2017 Disponible en ligne : www.atex-software.eu.
- [16] *Optimisation (mathématiques)* - Wikipedia. Disponible en ligne : [https://fr.wikipedia.org/wiki/Optimisation_\(math%C3%A9matiques\)](https://fr.wikipedia.org/wiki/Optimisation_(math%C3%A9matiques)).
- [17] Université Liège, *Communication graphique Bézier Et transformations affines*. Disponible en ligne : https://www.cgeo.uliege.be/communicationgraphique/Theorie/C3_Communication_graphique_blan.pdf.
- [18] TheAiLearner, *Perspective Transformation*. Disponible en ligne : <https://theailearn.com/tag/cv2-getperspectivetransform/>.
- [19] Pytorch, *Documentation - des deux fonctions get_perspective_coeffs et perspective*. Disponible en ligne : https://pytorch.org/vision/main/_modules/torchvision/transforms/functional.html#perspective.